AMERICAN UNIVERSITY OF BEIRUT

BATTERY MODELING FOR LIFE TIME ASSESSMENT

by
AMAL MOHAMMAD ASAAD

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
April 2017

AMERICAN UNIVERSITY OF BEIRUT

BATTERY MODELING FOR LIFE TIME ASSESSMENT

by

AMAL MOHAMMAD ASAAD

Approved by:

S. Karaki

Prof. Sami Karaki; Professor — Advisor
Department of Electrical and Computer Engineering

Ali Chehab

Prof. Ali Chehab; Professor — Committee Member
Department of Electrical and Computer Engineering

Prof. Rabih Jabr; Professor — Committee Member
Department of Electrical and Computer Engineering

Date of project presentation: April 25th, 2017

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: ___Asaad______________Amal_____________Mohammad_______
Last First Middle

☑ Master's Thesis ◯ Master's Project ◯ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after:

One ____ year from the date of submission of my thesis, dissertation, or project.

Two ____ years from the date of submission of my thesis, dissertation, or project.

Three ____ years from the date of submission of my thesis, dissertation, or project.

____

Signature Amal Date 2nd May 2017

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor prof. Sami Karaki for his continuous support and patience throughout my graduate study. Also he showed me a great motivation, enthusiasm and immense knowledge. His guidance helped me accomplish and writing this thesis. I couldn't have imagined having a better advisor and mentor for my graduate study.

Also I would like to thank the rest of my thesis committee: Prof. Rabih Jabr and Prof. Ali Chehab for their encouragement.

My Sincere thanks to Mr. Ali Kaafrani for his effort in updating the necessary software through the experiments, and offering help when needed any time without hesitation or delay.

Also a sincere thanks to MR. Mohamad Hussieni for his great effort in helping developing the C code, that is essential to drive the experiment and being there to help any time with high enthusiasm.

I want to express my sincere gratitude to all friends who encouraged me and stand by me through my hard days.

Also I would like to thank Mr. Fouad Shihab for supplying me all facilities to work in the laboratory with ease and comfort.

At the end, my eternal gratitude to my mother and sister for helping me, supporting me and never let me down and my kids Hadi and Dani because they are the real motive to go on and keep moving.

AN ABSTRACT OF THE THESIS OF

Amal Mohammad Asaad for Master of Engineering
Major: Power Engineering

Title: Battery modeling for life time assessment

Batteries have played a crucial role in power systems of different sizes, ranging from small portable electronic devices, to hybrid electric vehicle (HEV), as well as storage devices in renewable systems. The two main challenges of renewable energy sources are their intermittency and unpredictability, which are two drawbacks that can be mitigated by the usage of battery storage in the system. Many power engineers continuously thrive to understand the performance of batteries and their ageing process by modeling. The uncertainty of the expected lifetime of a battery has a significant impact on the cost of a given project as well as its effectiveness. This uncertainty in the life time of a battery can be limited by modeling it using circuit parameters and identifying how these parameters vary over time, which is the essential focus of this thesis. The ageing process of the battery is simulated in an experimental setup that charges and discharges the battery in a cyclic manner through the use of a programmable power supply and an electronic load. Three different approaches were attempted to estimate the battery lifetime. The first relies on using the total energy supplied by a battery through all the discharge cycles of the experiment as a reference for the energy supplied, obtained by integration of delivered power over time while the battery is being used. The second approach relies on discharging the battery for 20 minutes while measuring its terminal voltage; if it drops to below 1.75V per cell then the battery is to be replaced. The third method is based on the use of electrochemical impedance spectroscopy (EIS) to measure the variation in internal model parameters as the battery is cycled and thus aged. The models used were of the single and double Randle cell which consists of a resistance in series with one or two polarization circuits formed by one or more resistance in parallel with one or more double layer capacitance. Other models based on an infinite Warburg impedance element as well as a finite Warburg impedance model were also used. The Shepherd's model for discharging was also used to investigate how battery ageing affects the parameters of this model. Three lead-acid battery samples were tested through cycling while measuring the discharge time, the terminal voltage and current. An EIS device was used to measure the real and imaginary battery impedance using a frequency range of 1 to 1000 Hz at different stages of the cycling process from which the internal battery parameters were estimated. We finally used an Artificial Neural Network to relate these parameters to the total energy delivered through its life time which was considered to represent the age of the battery.

CONTENTS

Appendix

ILLUSTRATIONS

Tables

CHAPTER I

INTRODUCTION

Electrical energy is considered the most important form of power since it can be easily applied and converted into light, heat or any other form. The main problem is that electrical energy is hard to be stored in large quantities. Capacitors store it directly but with limited quantities, and its storage in larger quantities requires it to be changed into another form. In batteries, the energy of chemical compounds acts as a storage medium. Batteries are devices that can store the electrical power in the form of chemical energy and release it by internal chemical reactions when it is needed. Rechargeable batteries are electrochemical cells that convert the stored chemical energy into electrical form by a discharging process and vice versa by a charging process.

Batteries have played a crucial role recently in many power systems, for example they are used as power sources for portable electronic devices. Nowadays the trend is to use batteries as an economically and environmentally friendly power sources in hybrid electric vehicle (HEV) as well as storage devices in HEV's and renewable systems. Batteries do not have only an impact on the system operation and performance, but also they greatly affect the life cycle cost of a specific power plant. The two main challenges of renewable energy sources are their intermittency as well as their unpredictability. These two challenges can be faced by the usage of batteries in the system.

Power system designers in general agree that one of the essential weak links in the long-term operation of renewable rural-based energy framework is batteries due to their limited life and high cost. The construction of the electrodes of a battery and the electrolyte sets are the main ways to distinguish their structure. Many power engineers continuously thrive to improve the performance of a battery as well as its ageing process by modeling it. As it was mentioned, the uncertainty of the expected lifetime of the battery has a tremendous impact on the cost of the project as well as its effectiveness. The uncertainty can be limited by modeling it because this will help expect its life time more accurately and possibly help extend its efficient time. One method of modeling batteries is by using circuit parameters and identifying how these parameters vary over time as it is utilized.

Battery models focus on three different characteristics. The first and commonly used model is a simple performance model that depends mainly on modeling the state of charge of the battery. The second type is circuit model with parameters that is used to predict the terminal voltage and current and hence used for elaborate modeling of the battery operation and its management system to provide detailed calculations of the losses of the battery. The last type is the lifetime model that is used for assessing the impact of operating criteria on the expected life time of the battery. One of the widely used models uses the Shepherd that describes the relation of the output voltage with the state of charge and current throughput. The parameters in the Shepherd's equation can be related to the physical or chemical attributes of the battery which are different for the different types like internal resistance, open circuit voltage, discharge current and the state of charge [1].

Scientists investigated many different types of batteries, among them, lithium-ion and lead-acid. Many types of life time models exist but the main types are post-processing models, and performance degradation models. The performance model consists of two parts, the charge transfer and the battery voltage. The charge transfer is in fact the state of charge (SOC) which is evaluated by evaluating the time integral of the input current. The battery voltage is the voltage that is calculated using Shepherd equation. The battery degradation model is related to the continuously charging and discharging process, and how this process affects the model parameters. Another factor that affects the battery degradation is temperature. With higher temperature, the deterioration will be faster and so the energy that can be extracted from the battery will be affected. At higher temperature, a battery capacity will be higher compared to the same at lower temperature, but on the expense of its life time. For example a Lithium-Ion battery may loss about 10% of its capacity after 1000 charging cycles, under $25^{\circ}C$, of course the amount of capacity loss will be smaller after 1000 charging cycles at $40^{\circ}C$.

In this thesis, different approaches were followed based on an experimental setup to cycle the battery through charging and discharging and monitoring the voltage and current throughout the experiment. The first approach is based on using the electrochemical impedance spectroscopy (EIS) to determine battery parameters applying different models. The second approach based on the energy integration method where the total energy delivered by the battery is calculated using the discharge curves. The third approach based on carrying out a 20 minutes discharge test and observes the voltage across the battery terminals. The fourth approach is combining the first and second approaches using the Artificial Neural Network where the energy delivered from the battery is considered as an interpretation of the battery's life time.

Since the thesis based on experimental results, a full description of the experiment setup and description of the software that drive the instruments that charges and discharges the battery will be explained thoroughly. Then discussion of the sample output will follow. After that the results on models will be considered to determine which model is the best fit to the experimental data received from the experiment. Finally results based on the data will be promoted and analyzed to make a proper assessment of the life time of the battery based on the variation of its internal parameters.

CHAPTER II
LITERATURE REVIEW

Zeitouny et al [2], tried to model a typical battery using electrical circuit models that best fits its operational behavior. She used the least square estimation method to derive the parameters of the model design, which minimizes the sum of squares of errors between measured impedance values and those values given by the model. She used the impedance spectroscopy experimental means to obtain the values of the impedances measured. The experiments were applied to Lithium-iron as well as lead-acid batteries. Using the graphical user interface connected to computer to monitor the changes in the values of real and imaginary impedance values as the state of charge and frequency change. She applied the experiment to a single- time constant Randle cell, Two Randle cell, and a circuit using a finite Warburg to model the batteries. To make the behavior of the model tractable, an algorithm as well as an experimental setup is done simultaneously. She found out that the single Randle cell model didn't in fact represent the Lead-acid battery since the measured and the theoretical impedance values didn't match, while infinite Warburg impedance model shows a perfect match (from 0.1 HZ to 1 KHZ) specially at low frequencies. For Lithium-ion battery, also the single Randle cell failed to model the variations of the impedance values as the frequency and state of charge change. The Two Randle cell gave best representation of the behavior of the Lithium-ion battery (from 1HZ to 1 KHZ).

Tremblay et al [1] represented an easy – to – use battery model that was applied to dynamic simulation software. It used the battery state of charge only as a state variable in a trial to avoid the algebraic loop problem. This method is used to extract the model parameters and to approximate its internal resistance. The model is then validated by mounting the results with the manufacturer's discharge curves. Finally the battery model is included in the SimPowerSystems (SPS) simulation software. According to Tremblay et al, it is important to develop an accurate battery model which can easily be used with simulators of power systems and on- board power electronic systems. Based only on the SOC, the model is chosen in order to reproduce the manufacturer's curves for four major types of battery chemistries. These types are: Lead- Acid, Lithium-ion (li-ion), Nickel-Cadmium (NiCad) and Nickel-Metal-Hybrid (NiMH). The paper is divided into three sections. In the first section, the proposed model parameters are described based on utilizing simple controlled voltage source in series with a constant resistance. And a method is described to show to extract the battery parameters from the manufacturer's discharge curves of the battery. The second section presented a discharge curves that are obtained by simulations and are validated with the manufacturer's datasheets. The third section presents an application of an integrated battery model to the SimPowerSystems (SPS) used in the complete simulation of an HEV power train based on several assumptions. The validation approach is accomplished by comparing directly (by superposition) the obtained discharge curves using the model with those of the manufacturers. The model then integrated in the MATLAB-Simulink Sim- Power Systems library, and then is used in the SPS simulation of a complete HEV power train. The proposed model is simple and requires few parameters (only three points on the discharge curve which are defined as: the fully charged voltage, the end of the exponential zone, and the end of the nominal zone) and accurately represents the discharge curves of the manufacturers. The results obtained after applying the model show the possibility of using this battery model to represent transient states.

Zhang and Chow [6] proposed that it is important to find an accurate battery model that describes the precise battery state, such as the state of charge, the state of health and the state of function, which determines the power capability of the battery, for an optimal usage, since the PHEV applications require a battery model which can accurately simulate and predict the battery performance under different dynamic loads, environmental and battery conditions. He found out that such model prolong the battery life, and enable vehicle to grid and vehicle to home applications. To do that he proposed that such model should precisely describe the nonlinear I – V battery performance by examining the battery's hysteresis effect and the battery relaxation effect which are the corner stone of Zhang paper. The relaxation effect is simply the measure of time for the terminal voltage that the battery requires to relax after charging and discharging, to the new steady state value. For a certain PHEV applications an appropriate battery model is selected and this is done by formulating a multi-objective optimization problem balancing between the model accuracy and the computational complexity within the constraints set by the minimum acceptable accuracy and maximum allowable computational complexity time. An equivalent circuit based on the Thevenin's theorem is used as the basis for the battery model to provide an accurate battery I-V performance. Zhang used a series connected RC parallel circuits to model the battery's relaxation effect. The circuit is composed of two parts; the left part describes how the battery SOC varies with the current. The battery capacity is expressed by C_{Capacity}. The right part in turn describes the relation between the load current and the battery terminal voltage. Root mean square percentage error (RMSPE) and maximum percentage error (MPE) are used to quantify the model accuracy for models containing RC circuits ranging from one to five. Zhang found that the more RC circuits used increases the model accuracy, but at the same time increases the model's computational complexity. He formulated the model parameter determination as a typical least squares fitting problem and used the Gauss-Newton algorithm to solve the problem. Formulating a multi-objective optimization problem with two objectives: to maximize the model accuracy and to minimize the model complexity. Also two constraints had to be defined: the accuracy required and the maximum complexity allowed. By maximizing the modeling accuracy while minimizing the modeling complexity, the optimization problem tries to find out the optimal solution.

Li and Ke [4] presented a comparison study of mathematical and circuit-oriented battery models applied to lead- acid batteries, since these batteries are used for large power storage applications. This study shows how mathematical and circuit oriented battery models are developed to represent the batteries electrochemical properties. And then the relation will be analyzed. He showed that fundamental battery electrochemical relationship is explicitly built into the battery mathematical model but not directly in the circuit – oriented one. The mathematical battery models are mainly developed based on Shepherd's relation to predict system-level behavior, such as battery runtime, efficiency, or capacity. The Circuit –oriented models are electrical equivalent models which use the voltage sources, resistors, and capacitors combinations. There have been many circuit-oriented battery models. This paper presents this comparison study with a focus on the lead- acid batteries. For mathematical battery model, and based on Shepherd relation the fundamental voltage-current relation with the SOC is built into the model. While for the circuit –oriented battery model, the effect of the SOC to the circuit parameters is not directly tractable. There are several distinctions between the two approaches but the main difference is that regarding model extraction parameters of a mathematical battery model obtained by solving a system of equations or utilizing regression techniques. Taking into considerations that the parameter extraction of runtime- based circuit- oriented battery models requires more expensive computational power.

Low Wen Yao et al [5] developed an electrical battery model in MATLAB/Simulink. Where lithium Ferro Phosphate battery is presented. This model is claimed to be capable of simulating current- voltage performance in a highly accurate manner. He built his model based on Shepherd equation, but unable to characterize the nonlinear current-voltage performance of battery. So a new battery model based on the equivalent circuit model should be applied to MATLAB/Simulink to obtain a more accurate simulation results. Low found out the equivalent circuit model for battery developed in MATLAB/Simulink and the parameters of the battery model were determined using an experimental result. The model circuit is a dynamic equivalent circuit model consists of a dc voltage source, a series resistance and two RC parallel circuits. Successfully representing the open circuit voltage (OCV) using DC voltage source, the series resistance is used to represent the internal dc resistance and RC parallel networks (R1, C1, R2, C2) are used to represent the transient response voltage. The comparison results come up to show that the voltage curves of the simulation and the experimental results are perfectly matched. The RMS error of voltage in random load test is 20.32mV, which is 0.635% to the nominal voltage of the battery. This confirms that the developed model can perform accurately in random load conditions. So he concluded that an accurate MATLAB/Simulink battery model is recommended to characterize the dynamic characteristics of the Lithium –ion battery.

Layadi et al [9] stated that the battery lifespan depends mainly on the number of cycles and the depth of discharge (DOD). But he added that in renewable hybrid power systems, the charging and discharging cycles are random and not regular. A relevant way for aging lead-acid batteries connected to a standalone multi source renewable system has been developed. It depends on the Rain Flow method for counting cycles and considers instantaneous DOD and average temperature. In this paper Toufic settled a method for classifying the number of cycles for each functioning year according to the DOD. Based on these data, the battery degradation rate is being estimated and so it is possible to come up with conclusions concerning battery life span. He promoted the method to be applicable to a multi power-system simulated dynamically under MATLAB/Simulink. According to Layadi et al, It should be taken into account several inputs and elements such as sun radiation, wind speed, load profile, photovoltaic generator, wind turbine and diesel generator to have results with good accuracy. Also he proved that the DOD as well as the nominal battery capacity are factors that affect the battery lifespan. In fact he investigated two models, the first represents the variation of the number of cycles according to temperature, and the second model represents the variation of the number of cycles according to various depths of discharge DOD. He supposed that we should simulate the battery by different values of maximum DOD if we want to successfully evaluate the aging of the battery bank. Then he presented a

Rain Flow algorithm he had used to extract the cycles of the battery operation. Where this algorithm is based on the extraction of cycles from a signal which represents the degradation of the system.

Stephen Buller et al [23] employed the electrochemical impedance spectroscopy to find the equivalent battery parameters for a super capacitors and Lithium-ion batteries. He used the Lithium –ion and super capacitors to present the nonlinear and lumped-element-equivalent circuit models that give the accuracy requirements for energy storage devices models simulations. The method of EIS is used and the simulation results are compared to test-bench data. Perfect matching of simulation results and measured voltage data. Due to the versatility of his approach (impedance –based modeling), then it can promoted to the fuel oil stacks in the future. And of course will give a chance for the combined simulation of different energy storage devices on a purpose to evaluate the new storage-hybridization concepts.

Daowd et al [3] proposed a new method to estimate the parameters of the battery based on MATLAB/Simulink estimation tool for three famous battery models; the partnership for a New Generation of vehicles (PNGV), Thevenin and second order battery model. He used Lithium polymer (Li-Po) 12Ah, 3.7V battery to be tested by a specific standard tests under different states of charge (SOC), temperatures and current rates. Three model parameters are estimated in these conditions. To acquire an acceptable results, two steps has to be aware of, the first step is applying a standard tests for estimating the parameters of the battery. The second step is to use an accurate method to estimate the battery parameters with minimal errors. So he followed a new parameter estimation method using MATLAB/Simulink parameter estimation tool, which is easy, fast processing time, useful, powerful tool and it is applicable for any battery model. The validity of the Simulink for any model is a spectacular advantage which makes it a special tool in parameter estimation as well as model modification, also it requires less processing time, easy to implement, and a powerful tool that gives accurate results. The second order battery model gives the best response with the least error into the battery simulation modeling. Also the best response can be noticed during the validation simulations. Thevenin battery model in turn, gives the maximum voltage error while the PNGV battery model gives an error that is intermediate between Thevenin and the second order model[3].

Jongerder and Haverkort [7] explained the effect of nonlinear physical effects in the battery. The life time depends on the usage pattern beside the rate of consumption of the device. He demonstrated that during the periods of high energy consumption, the effective capacity degrades; hence the lifetime will be reduced. However, during periods without energy consumption, the battery can recover some of its lost capacity, and consequently the lifetime will be extended. They investigated different approaches that have been utilized to model batteries. They studied their properties, starting from very detailed electro-chemical models to high level stochastic models. They gave a deep insight of the electrochemical basics of batteries and define what so called ’recovery effect’, which is the recovery of the battery capacity lost during the periods of high discharge when the discharge rate is lowered to a certain extent. Also they define the electrochemical models which are based on chemical process that takes place in the battery.

They used Dualfoil, the FORTRAN program to simulate the Lithium-ion batteries using an electrochemical model. The program deduces how the battery properties change overtime for the load profile set by the user. From the output data, they proposed it is possible to come up with the battery life time. Beside the load profile, the user has to set over 50 battery related parameters, like, the thickness of the electrodes, the initial salt concentration in the electrolyte and the overall heat capacity. They investigated the electrical circuit models. They concluded that although the electrical circuit models are simpler and computationally less expensive than the electrochemical models, they still require massive efforts to construct the electrical circuit models beside they are less accurate. Further they defined the analytical model that describes the battery at a higher level of abstraction than electrochemical and electrical circuit models. At the end, they defined the Peukert's law which is the simplest model used to predict the life time of the battery by relating the nonlinear relationship of the lifetime of the battery and the discharge rate without considering the recovery effect.

Unterrieder et al [8] proposed a scheme that presents a battery open-circuit voltage extrapolation method that uses a modified version of the approximate least squares estimation scheme for the estimation process. The proposed project is applied to battery state-of-charge estimation. By predicting the battery's electromotive force, the suggested approach allows for a highly improved re-initialization of the Coulomb counting based state-of-charge estimation method. The proposed methodology makes use of the EMF- SOC table based re-calibration method.

Miriam et al [11] used the Artificial Neural Network with back propagation to predict the life time and the behavior of lead acid battery as a function of less discharge time, less energy and less working hours. The experimental training data set like discharge voltage and time were used to predict the unknown outcomes. It proved that the predicted values were close enough to the practical results. They showed an acceptable performance and capabilities. They concluded that they can be considered as a leading resolution concerning maintenance and a guide decision to replace the used battery by learning. They performed experiments concerning the prediction of a battery with a floating service and daily discharge, prediction of the battery life with endurance cycle (life cycle test) and the prediction of the discharge manner at high discharge rates and cold cranking at low temperature. They concluded that the results were good without a need for many experimental data.

Thesis contribution:

The main focus of this thesis is to determine a method or approach to estimate the battery life-time or to clearly answer the question: should the battery be replaced or not?

Three different approaches were attempted to estimate the battery life time. The first relies on using the total energy supplied by a battery through all the discharge cycles of the experiment as a reference for the energy supplied, obtained by integration of delivered power over time while the battery is being used. The second approach relies on discharging the battery for 20 minutes while measuring its terminal voltage; if it drops to below 1.75V per cell then the battery is to be replaced. The third method is based on the use of electrochemical impedance spectroscopy (EIS) to measure the variation in internal model parameters as the battery is cycled and thus aged. The models used were of the single and double Randle cell which consists of a resistance in series with one or two polarization circuits formed by one or more resistance in parallel with one or more double layer capacitance. Other models based on an infinite Warburg impedance element as well as a finite Warburg impedance model were also used. The Shepherd's model for discharging was also used to investigate how battery ageing affects the parameters of this model. Three lead-acid battery samples were tested through cycling while measuring the discharge time, the terminal voltage and current. An EIS device was used to measure the real and imaginary battery impedance using a frequency range of 1 to 1000 Hz at different stages of the cycling process from which the internal battery parameters were estimated. We finally used an Artificial Neural Network to relate these parameters to the total energy delivered through its life time which was considered to represent the age of the battery.

CHAPTER III

STATE OF HEALTH AND STATE OF CHARGE

To understand what the battery state of health means, it is better to understand the electro-chemical processes that the battery undergoes which controls its performance and life degradation. Battery capacity can be defined as the total charge or energy delivered by a fully charged battery before depletion. For a certain temperature and discharge current, the rated capacity is given by amp-hours (Ah). The available capacity depends mainly on the discharge current and the temperature, when they vary its actual available value will not be the same. Before going on the definition of the battery state of health, its state of charge (SOC) and battery aging, a closer look about the chemistry of the battery will be beneficial understanding the behavior of the battery and its characteristics.

A. Battery chemistry background

The chemical process that occurs in the cell is responsible for the dynamic electrical behavior of the battery during charging and discharging operation. Understanding the reactions that occur in the battery's cell gives us a great insight of its functionality.

In the active material bonded to the positive and negative metallic electrodes, the battery cell stores the electrochemical energy. When an external circuit is connected to the battery electrodes, chemical compositions change and consequently, electrons will have the mobility from one active material to the other. The shuttling ions between the active materials cause the electrolyte to participate in the reaction. These electrochemical reactions causes the battery to deliver energy to the load connected to it during the discharge process and to absorb the electrical energy from the connected electrical source when it is in the charging state.

In lead-acid batteries, the positive active material is a paste of lead dioxide (pbo2), the negative material is a porous sponge lead (pb), and the electrolyte is an aqueous solution of sulfuric acid (H2SO4)[11].

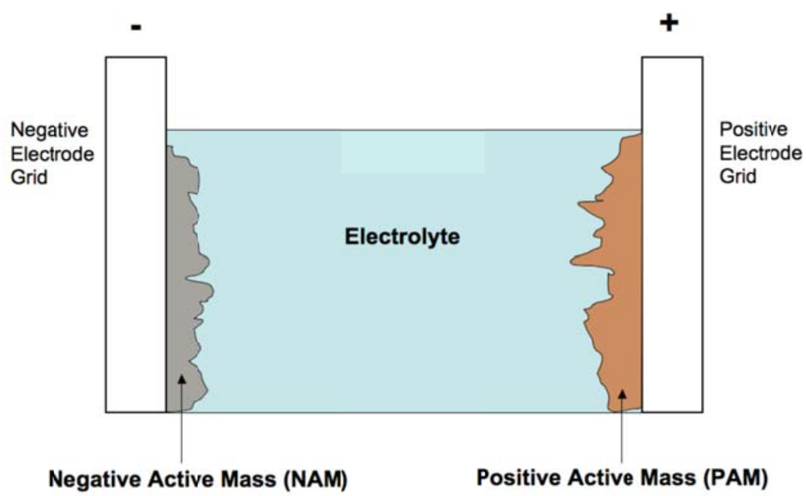


Figure 1.The chemical structure of Lead-acid battery

The entire process at either electrode begins with the electrochemical dissolution reaction, which involves electron transfer, followed by the precipitation of solid lead-sulfate.

The overall discharge reaction creates non-conducting solid PbSO4 on both electrodes by breaking down sulfuric acid and producing water at the positive electrode, making the electrolyte much more dilute than it was originally. As one might imagine, as long as the discharge reactions proceed and the SOC is depleted, causing an increase in the battery’s internal resistance that is a result of increased current densities within the active mass and decreased active surface area. Knowing the reactions and reactants that must be present at each active mass surface for current to flow is very important in understanding the dependence of battery performance on operating conditions. Furthermore, the morphological structure and availability of the active materials themselves will also play a large role in battery electrical behavior. Measurements of voltage and open circuit voltage during discharge will help determining the State of health (SOH) and the state of charge (SOC) of the battery.

In lead-acid batteries, the electrolyte is diluted and broken down to lead-sulfate and water as the SOC depleted. In the following sections we are going to define the state of health and the state of charge and present the method used to measure them.

B. Definition of state of health (SOH)

The state of health is a measure that reflects the condition of the battery or the performance of the battery compared with the fresh unused one. In another way it is a figure of the quality of its condition compared to the condition of an ideal one. It is measured in percentage. i.e. that a battery's state of health is equal to 100% when it is newly produced or manufactured, and this percentage will decrease over time and usage. The state of health is an important parameter of the battery to measure its remaining life time, and to change it before its total failure.

The performance or the "health" of the battery continuously deteriorate due to the physical and chemical reactions that take place during the usage of the battery till it ages and no longer usable. The SOH is acting as an indication of the condition of the battery relative to the fresh one.

There is no obvious method to measure the SOH since it is an estimation rather than measurement according to people perspectives. SOH tracking techniques are in their infancy, the battery management system defines the SOH as the ability of the battery cell to store energy, deliver, receive power and high currents and hook up charge over long periods relative to its nominal capabilities.

1. Parameters

Unlike the state of charge (SOC) that can be evaluated by measuring the actual existing charge in the battery, measurement of the state of health is a subjective reliance in that engineers derive it from the variety of different measurable battery parameters. Since the SOH applies only to batteries that started to age either due storage or entered service. The SOH is determined by the test equipment's or by the users themselves via determining some parameters. The battery designer management system may use one or a combination of the following parameters to derive a value for the state of health since SOH doesn't correspond to any physical interpretation, these parameters are:

1- Internal resistance

2- Capacity.

3- Voltage.

4- Self- discharge.

5- The ability to accept a charge.

6- Cyclic number (number of charge- discharge cycles).

The most important parameter of the battery to estimate its state of health is the capacity which can be determined by different methods, but the direct evaluation of the capacity is time consuming beside it is very difficult to find out the value of the capacity when the battery is operating. So we can measure any of the above parameters which have a good correlation to the capacity, but on a condition that the measured parameters at time t should be linked to the capacity at that time t. In VRLA (valve regulated Lead acid) batteries, measurement of the battery impedance proved to have the best results for determination of the state of health for VRLA batteries.

2. SOH applications

Usually the battery management system estimates the SOH of the operating battery [14]. This estimation is then compared to what so called the SOH threshold of the given implementation. Comparing these two values gives a clear insight whether the selected battery is suitable for this usage, and also lifetime assessment of the battery is then possible. So we can conclude that the purpose of the state of health is to give an insight of the expected performance of the battery when put in operation and to provide an indication of the life time consumed and the useful lifetime remaining before the battery ages. Engineers can use SOH to contemplate problems concerning fault diagnosis or replacement of recent plan. SOH mainly is a long term battery changes tracking phenomena.

C. State of Charge

The state of charge (SOC) indicates the amount of charge that the battery can deliver with respect to its nominal capacity. Measuring the amount of energy left in the battery relative to the total energy that it had when fully charged, gives an indication of the time that the battery will perform a certain application before recharging. It is a short term capability measure of the battery. A fully charged battery will have a 100% SOC while the fully discharged one will have a 0% SOC. The SOC is usually defined as the available capacity expressed as a percentage of a certain reference. Sometimes this reference could be the rated capacity of the battery but usually the reference is chosen to be its current.

Engineers prefer to consider the new cell rated capacity as a reference rather than the current capacity of the cell. The reason is that the cell capacity is affected when the cell ages; also it is affected by the temperature and the discharge rate since temperature and discharge rate reduces its effective capacity. Battery's SOC is very important parameter in the modeling and diagnostic perspectives. From a modeling perspective the dynamic features of the battery changes with the SOC, and if the battery is operated or stored at low SOC, it will cause an irreversible damage to the battery in the diagnostic perspective.

1. Determining the state of charge

SOC can be estimated by many methods. Most consider the measurement of a suitable parameter that varies with the state of charge which is related to the chemistry changes that happen in the battery cell due to charging and discharging the battery.

The direct measurement is the easiest method for measuring the SOC, but this would happen if the battery is discharged at a constant current rate. It is known that the battery charge is a result of the current multiplied by the time required for discharging. Two draw backs are in fact facing this method. First the discharge current in the practical cases is not constant; in fact the current is continuously reduced as the battery discharges in a non-linear manner. For this reason any measurement apparatus should be able to integrate the current over the discharging time. Second, this method rely on the discharging the battery to recognize the amount of charge it captured. But in real applications it is preferred to know this amount of charge without the need to discharge the battery. Further due to the losses caused by charge-discharge cycles, the battery will deliver less amount of charge than the amount it receives when charging. Other methods for SOC measurements are

a. Specific gravity (SG) measurements for SOC

It is an accustomed way for recognizing the charge condition of the lead acid battery depending on finding the variations of the weight of the active chemicals. As we explained earlier, as the discharge process is going on the electrolyte (Sulphric acid) is consumed and its concentration is reduced and consequently reduces the specific gravity of the electrolyte in proportion to SOC. Accordingly this can be used as a hint of the state of charge. Slow and awkward suction hydrometer type was used to measure the specific gravity. Now electronic sensors with digital measurements can continuously and directly give readings of the electrolyte specific gravity.

b. Voltage based estimation

This method is based on measuring the voltage of the battery to determine the SOC or the active remaining capacity of the battery. It has a disadvantage that is the lack of accuracy since it depends on the actual voltage, the rate of discharge, temperature, and the battery age so a compensation factor should be taken into consideration to achieve an acceptable accuracy.

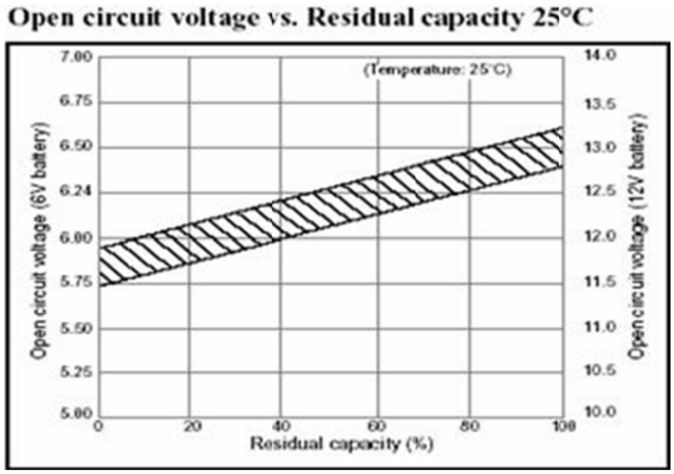


Figure 2. Lead acid battery

Figure 2 above shows the relation which is a direct proportion between the open circuit voltage and the available capacity at constant temperature and discharge rate for lead acid battery.

c. Estimation of SOC based on current

Coulombs is the unit of the electric charge which can be evaluated by integrating the total current delivered by the charge over the time of discharge. The remaining capacity of the battery cell can be calculated by the current received when charging and delivered when discharging the battery then, integrating over the total time period. Then accumulating the current over time will enable us to evaluate the charge in or out the cell. Then the SOC can be obtained by subtracting the total charge flowed from the charge in the fully charged battery. This method is the most simple and accurate method since it depends on the direct measure of the charge flow. It is called the “coulomb counting”.

But also it needs a compensation factors for accuracy.

For that reason we can use current sensing methods like:

1- Current shunt: it is a low ohmic value, series, sense and high precision resistor between the battery and the load. It is required to measure the voltage drop across it. But the main drawback of this method is the power loss in the current path and it may cause a battery temperature rise. Another thing it has a lack of accuracy for low currents use.
2- Hall Effect: to avoid this problem engineers use transducers but they are expensive and can't withstand high currents, further they are exposed to noise.
3- GMR: magneto resistive sensors are more accurate, higher sensitivity, higher temperature stability and provide higher signal level than the Hall Effect devices but they are more expensive.

We should note that the coulomb counting depends on the current flowing from the battery (discharge) to an external circuit but not self-discharge currents.

<u>d. Internal impedance measurements for estimating the state of charge</u>

Between the charge and discharge state, the composition of the active chemicals in the battery cell keep changing. And accordingly the cell impedance will change. So we can use the measurement of the cell internal impedance to determine the state of charge (SOC). This method has limitations due to the difficulties in measuring the impedance of the battery while using it, further, the internal impedance itself is temperature dependent.

CHAPTER IV

BATTERY MODELING

A. Electrochemical Impedance Spectroscopy (EIS)

Modeling the battery to better understand its behavior and estimate its life time directly requires deep study of the chemistry of the battery as we attempted to do in the last chapter. Alternatively, the chemical reactions taking place in the battery have an effect on the electrical parameters of a postulated model, which may be measured using a powerful tool known as electro-impedance spectroscopy (EIS) that has the merit to supply us with the accurate and free – error kinetic and mechanistic information in the study of corrosion, batteries, electroplating and electro-organic synthesis.

Electrochemical impedance is usually evaluated by applying an AC potential to an electrochemical cell and measure the current through the cell at different frequencies. The main feature of (EIS) is that one can simply represent the electrochemical cell by the usage of an electric model. When electrochemical reactions occurred at an electrode interface, it is identical to an electronic circuit or any equivalent circuit consisting of a combination of resistances and capacitors that are connected in series, parallel or series – parallel combination, since the chemical reactions with electrodes can obstruct the flow of the electrons that have resemblance to an electric circuit elements (resistors, capacitors and inductors) behavior. Also the (EIS) can signal the damage of the coating on the metal substrate. Once a particular model is adopted, the physical and chemical properties of the elements of the circuit can be correlated and hence, assume numerical values obtained by fitting the measured data to the model elements. [17]

For this reason, the theory of the electric circuit will be applied starting from the "Ohms Law". As we know the impedances of any electric circuit are R, ωL and $\frac{1}{\omega C}$. Where ω is the angular frequency given $\omega = 2\pi f$ where f is the frequency of the applied signal in hertz (Hz).

Nyquist and Bode plots are the most widely used data plots in the (EIS) since the frequency is shown on x-axis. Further, the Bode plots are logarithmic scale plots.

B. Equivalent Circuit Models

Several models are investigated in this chapter to monitor the behavior of the battery and try to predict its life time by monitoring the variation of its model's internal parameters using the method of charging-discharging (cycling). Here we will discuss each model, its concept, the theory, the rules and equations used to find its parameters. Here we present the models in increasing level of complexity and expect that each model will give a more accurate and robust results than the preceding one. First, I will start with the single Randle cell model that would be considered as the corner stone to the other models. Second, the double Randle cell will be analyzed and try to use it to figure out the parameters of the battery for its life time assessment. Then a model with a finite Warburg impedance element will be examined to find out if it really represents the measured data and the parameters of the battery. After that, a more generalized model, infinite Warburg impedance model is analyzed and examined to fit the data. Finally a Shepherd equation model for an open circuit voltage is examined and a modified Shepherd model is tried to fit our data and then monitor the variation of its internal parameters to predict the battery life time.

1. Single Randle Cell

The Single Randle Cell shown in Fig. 3 is considered the simplest model used to represent the battery's functionality. It consists of a resistance that represents the electrolyte uncompensated solution resistance (R_s). This series resistance represents the opposition to the current flow, since most electrochemical cells don't have a uniform distribution of currents through electrolytes. Calculating this series resistance is cumbersome because we should determine the current flow path and the electrolyte geometry. However, it is found by fitting the battery's model to experimental EIS data. R_s is connected in series with a parallel combination of RC circuit. RC is the time constant of Randle cell formed by what so called the polarization resistance R_p in parallel with C_p. To understand what R_p stands for, we should define the word "polarizing the electrode". It is referred to the potential of the electrode that is forced away from its open circuit value. When electrode polarization occurs, it causes a current flow by the electrochemical redox reactions that happened on the electrode surface; hence R_p is the charge transfer impedance. While C_p represents the double layer capacitance. The double layer capacitance is the layer that is formed when solution ions stick on the electrode surface. Electrodes charges are separated from these ions charges, this separation is of the order of angstroms, and as we know charges separated by an insulator form a capacitor. Many factors affect the value of the double layer capacitor, like, electrode potential, temperature, ionic concentrations, types of the ions, impurity, layers roughness and oxide layers. The time constant R_pC_p symbolize the time that the battery needs to change its state from transient to steady state. Single Randle cell is shown below.

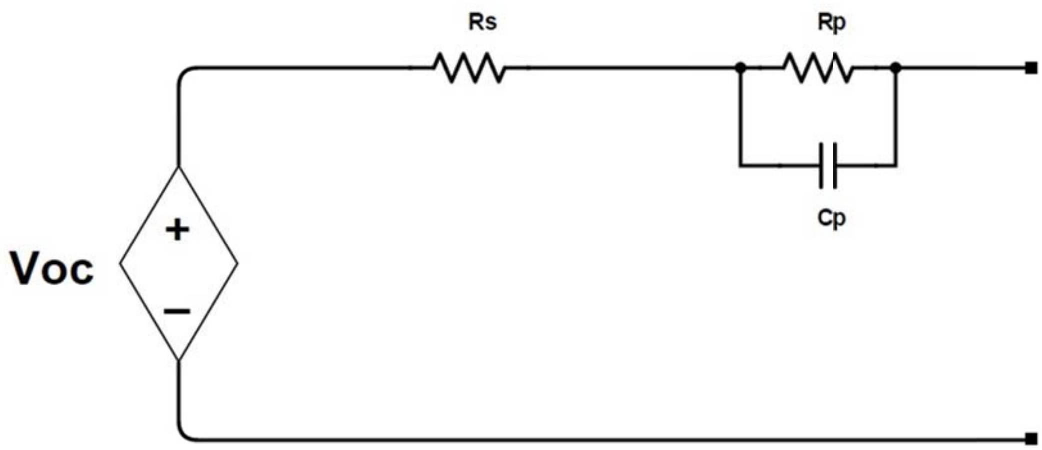


Figure 3. Single Randle Model

To find the equivalent impedance

$$Z_{eq} = R_s + \frac{R_P}{1+j\ \omega \times R_p \times C_p} \qquad (1)$$

This equivalent impedance is implemented in MATLAB and compared to its measured value to minimize the sum of squares of errors in order to extract the parameters of the battery, the series resistance, the time constant elements parallel resistance and the double layer capacitance. This code is found at the end of the thesis book. The limitation of Single Randle Cell is that the Nyquist plot of the model is always a semicircle shape whether the measurements are taken at low or high frequencies. So Nyquist is suitable for monitoring the effect of the resistive loads.

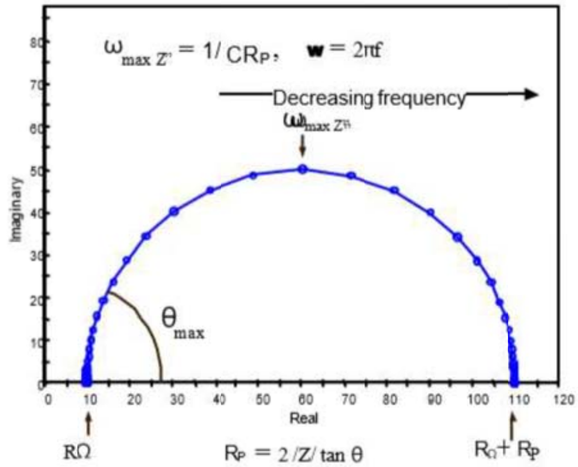


Figure 4. Nyquist plot for Single Randle Cell [14]

2. Double Randle Cell

This model has two time constants instead of one, these two time constants are used to represent the short- term transient and the long- term transient behaviors. Increasing the number of time constants increases the modeling accuracy but at the same time increases the modeling computational complexity [6]. Two time constants can track the I-V characteristic and predict the run time as well as the state of charge. The Double Randle cell circuit is as follows

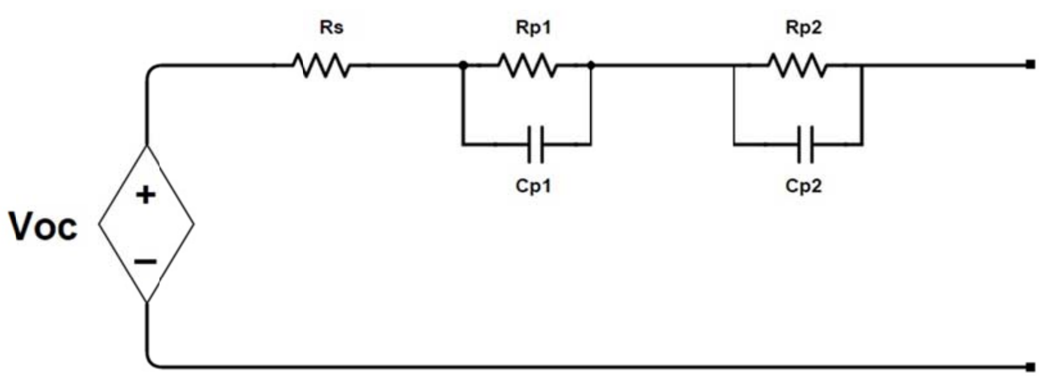


Figure 5. Two Randle Cell

And the equivalent impedance can be derived as follows:

$$\boldsymbol{Z_{eq} = R_s + \left(R_{p1} \left\| \left(\frac{1}{j\omega C_{p1}}\right)\right.\right) + \left(R_{p2} \left\| \left(\frac{1}{j\omega C_{p2}}\right)\right.\right)}$$

This leads us after simplifications and using series parallel rules to

$$Z_{eq} = R_s + \frac{R_{P1}}{1+j\ \omega \times R_{p1} \times C_{p1}} + \frac{R_{P2}}{1+j\ \omega \times R_{p2} \times C_{p2}} \qquad (2)$$

Double Randle Cell model proved to be suitable to model the Lithium- ion batteries; but lack the accuracy to model the lead acid batteries; hence it is poor in predicting the life time of lead acid batteries, however the model was applied as will be seen in the next chapters.

3. Warburg Impedance Based Models

The Warburg diffusion element is an equivalent circuit that characterizes the diffusion process that is associated with the charge transfer resistance and a double layer capacitance. It is related to the mass transfer of ions in the electrochemical system, and its presence can be justified if a linear relation of log |Z| versus log (ω) has a slope equal to (- $\frac{1}{2}$).

The circuit model of a single Randle's cell with a Warburg Impedance to represent diffusion is shown below in Fig. 6.

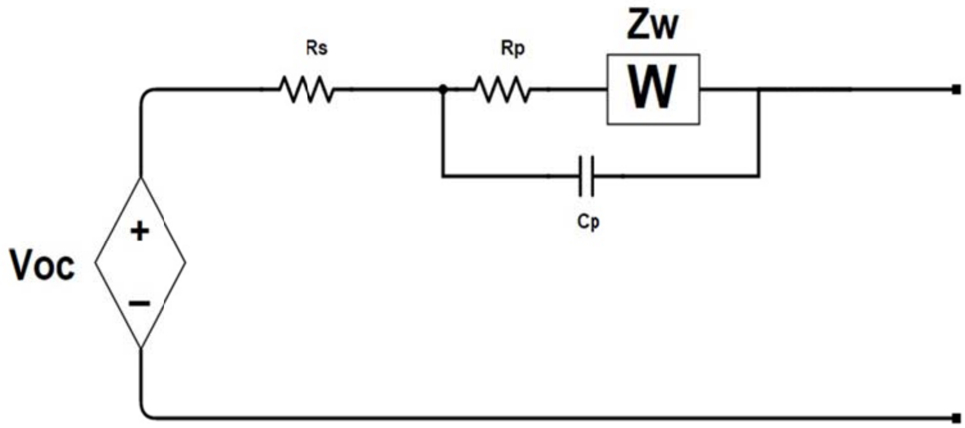


Figure 6. The Warburg impedance model

As defined, R_S is the series resistance, R_p is the charge transfer resistance and C_p is the double layer capacitance. Z_w Is the Warburg diffusion impedance, which depends on the frequency. At very high frequency, diffusion occurs in at a low rate causing a correspondingly low value of the Warburg impedance. While at low frequencies, the diffusion rate gets larger causing noticeable Warburg impedance. The reliance of the Warburg impedance is given by the following equation [2]:

$$Z_\omega = \frac{A_\omega}{\sqrt{\omega}} * (1-j) * tan\,h\left(\delta * \sqrt{\frac{j\omega}{D}}\right) \qquad (3)$$

A_ω: is the Warburg constant

δ: Nernst diffusion layer thickness.

D: average value of the diffusion coefficients of the diffusing species.

The above equation is valid for all frequency ranges and the model is then known as the finite Warburg impedance model. At high frequencies Z_ω can be approximated to:

$$Z_\omega = \frac{A_\omega}{\sqrt{\omega}} * (1 - j) \qquad (4)$$

Where “tanh’’ equals 1 at high frequencies [2]. The model would then be known as the infinite Warburg impedance model.

The above equation can model the lead-acid battery internal circuit but at low frequencies to an acceptable accuracy.

C. Shepherd’s equation

Shepherd derived an equation directly describes the electrochemical behavior of the battery using several terms like, the terminal voltage, open circuit voltage, internal resistance, discharge current and the state of charge. This model is applied to charging or discharging states as well. The main problem with Shepherd is that it causes an algebraic loop problem in the closed loop simulations [2]. The Shepherd’s model describes the voltage – current characteristics. It is the preferred model for the photovoltaic (PV)-systems because it offers a high precision, further fewer parameters are involved with Shepherd model than other parameters. The parameters are extracted from experimental data analysis. Shepherd’s equation is expressed as follows [20]:

$$U_{cell} = U_{oc} - g_c (1 - F) + \rho_c \frac{I}{C_N} + \frac{\rho_c * M_c * I}{C_N} \frac{F}{C_c - F}$$

(For I >0, battery is charging)…… (5)

$$U_{cell} = U_{od} - g_d (1 - F) + \rho_d \frac{I}{C_N} + \frac{\rho_d * M_d * I}{C_N} \frac{F}{C_d - F}$$

(For I <0, battery is discharging)….. (6)

The above equations contain four terms:

The first term is the open circuit voltage which balances the cell voltage after full charge and sufficient rest period $\boldsymbol{U_{oc}}$.

The second term is related to the state of charge. The third one represents the ohmic losses that are proportional to the current. The fourth term characterizes the charge factor over voltage that increases when the battery is nearly empty or fully charged. [2]

In the above two equations we had:

$\boldsymbol{U_{cell}}$**:** The terminal voltage of the battery in [V].

$\boldsymbol{F}$: The normalized SOC and it equals (1 – H), obtained by integrating the effective current then dividing it with the nominal capacity.

$\boldsymbol{I}$: discharge current (I <=0) or charge current (I >0) in [A].

$\boldsymbol{U_{od}}$ **,** $\boldsymbol{U_{oc}}$: the open circuit equilibrium voltage in [V]

$\boldsymbol{g_c}$ **,** $\boldsymbol{g_d}$: proportionality constant over the electrolyte in [V].

$\boldsymbol{\rho_c}$ **,** $\boldsymbol{\rho_d}$: The internal resistance parameter [ΩAh].

$\boldsymbol{C_N}$: The battery nominal capacity defined by the manufacturer [Ah].

$\mathbf{M_c}$: charge transfer over voltage coefficient.

$\mathbf{C_c}$, $\mathbf{C_d}$: the normalized capacity coefficient.

If we plot $\boldsymbol{U_{cell}}$ versus time of the experimental measured data versus the model we will note that the measured data trend has an exponential zone in the first region (Fig7). For that reason an exponential term is added so the above equation for discharging becomes:

$$U_{cell} = Ae^{-B(1-F)} + U_{od} - g_d\,(1-F) + \rho_d \frac{I}{C_N} + \frac{\rho_d * M_d * I}{C_N}\;\frac{F}{C_d - F} \quad \text{..... (7)}$$

And if we substitute H = 1 –F, where H is the depth of discharge. We get the following equation

$$U_{cell} = Ae^{-BH} + U_{od} - g_d H + \rho_d \frac{I}{C_N} + \frac{\rho_d * M_d * I}{C_N} \frac{1-H}{C_d-(1-H)} \text{..... (8)}$$

The resulting graph of the measured voltage versus the time of charge or discharge (in our case, the discharge time is considered) has the following pattern

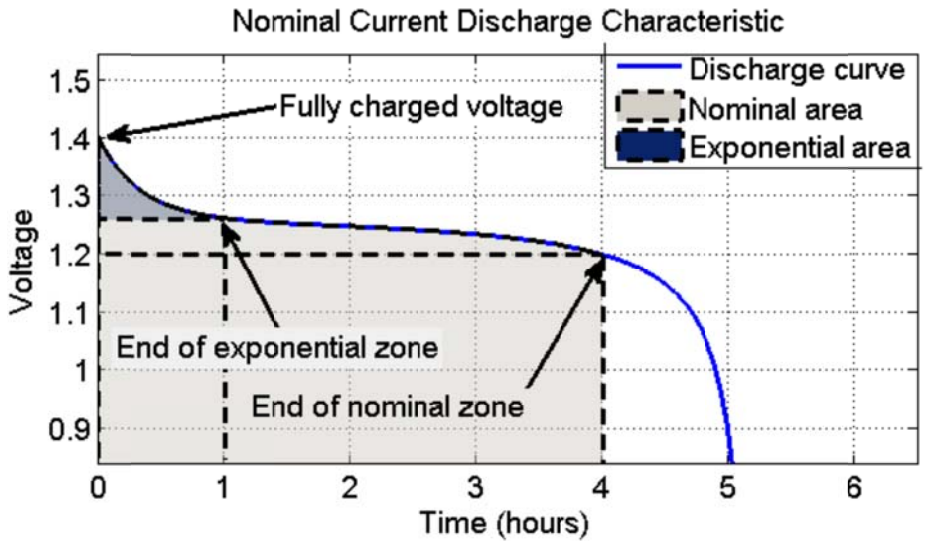


Figure 7. The voltage- time curve for Shepherd model

Plotting the discharge successive curves throughout cycling will result the following plot in Fig.8. We used this plot to approximate the total energy delivered by the battery when discharged at I_4 constant current in KWh. Fig.8 shows the discharge curves for voltage versus time. U_1 corresponds to the discharge voltage in the first cycle in the first loop (each consists of 20 cycles), U_2 is the first discharge voltage in the second loop i.e. cycle 21, while U_3 is the first discharge voltage in the third loop so it is the discharge voltage in cycle 41 and so on.

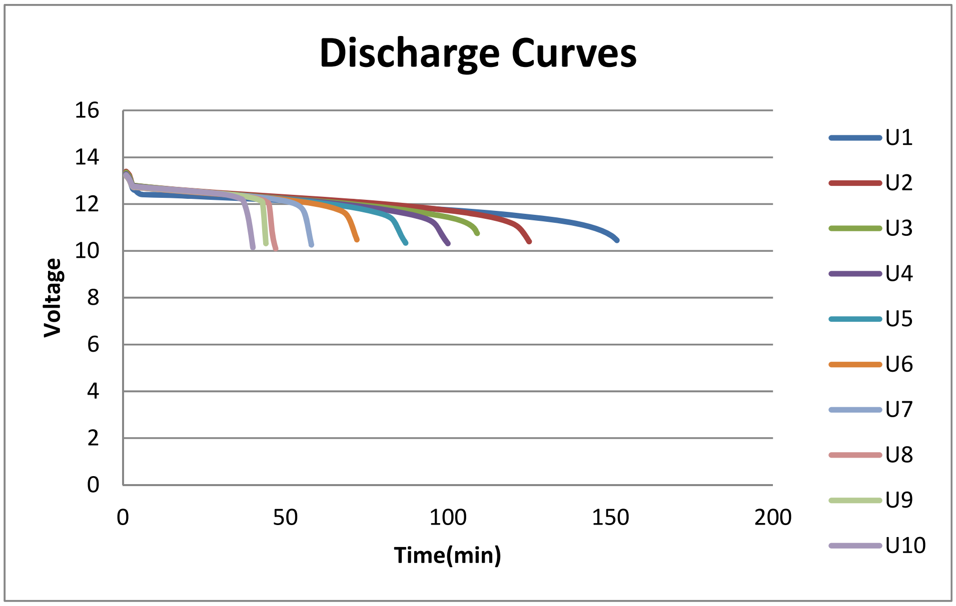


Figure 8. Discharge curve for one of the tested lead acid batteries

D. Methodology used for battery life time estimation

According to the above explanations of the models proposed, our strategy to estimate the life time will be based on three approaches:

1. Energy integration

Figure 9 below shows the battery life time detection using energy integration. The theory behind this is a reference measure of the energy delivered by a battery during the discharging process (E_0) over its lifetime may be estimated over the 200 cycles of charging-discharging process. As the battery is utilized, the energy delivered up to a point in time (E) can be evaluated by integrating the product VI over time as shown in Fig. 9. The ratio E/E0 may be considered as a measure of the lifetime of the battery in per-unit and can be converted into time by knowledge of the current and voltage. By taking into account the distribution of the current ($I \pm 3\sigma_I$) the life-time may be predicted with uncertainty limits.

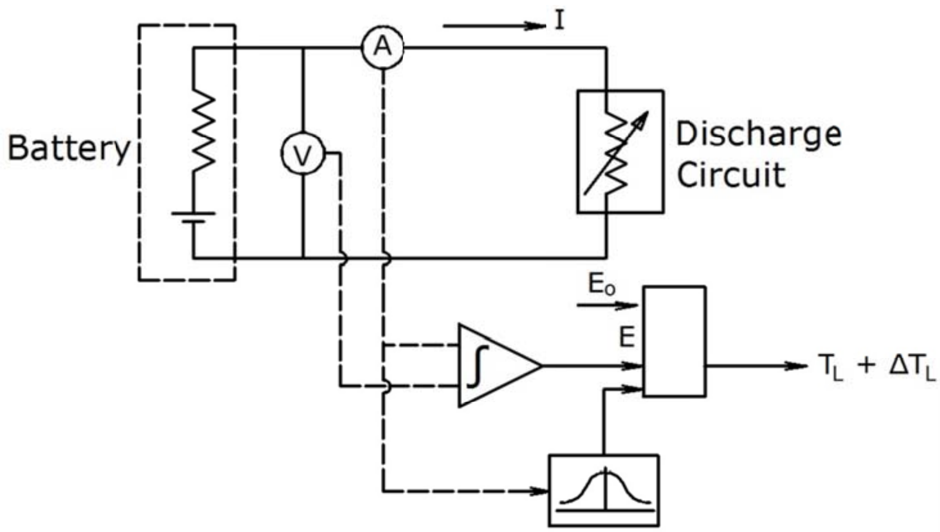


Figure 9. Battery life time detection using energy integration

Hence , recall Fig. 8, by using the discharge curves for the battery, we can estimate the energy delivered between U1 (first discharge) curve and U2 the (21st discharge), and then similarly find the total energy delivered all over the 200 cycles (battery ages) and hence, determine what is the total amount of energy that we can get through its life time and constantly compare this to the amount of energy delivered up to a certain time and estimate the amount that can still be used in the future given the level of usage and its distribution. Full details of the method will be discussed with the results extracted from testing three lead acid batteries in the next chapter.

2. Energy discharge time

The second method is used to estimate the time when the battery needs to be replaced by monitoring the time it takes to discharge as it ages. Figure 8 clearly shows that as the battery ages it's time to completely discharge drops from 150 down to 20 minutes. This was used to propose the following test: when the time of discharging (20 minutes) the battery starting at 100% SOC causes the voltage to drop to below a certain specified voltage (in this case 10.5 V) then we conclude that the battery should be replaced. If not then the battery still can be used.

3. Artificial Neural Network for battery parameters

Artificial Neural Network (ANN) is an algorithm that was inspired from the biological human cortex. The multilayer neural network is a system made up of layers containing the neurons; neurons are connected to each other in different layers to form a network. Each neuron is called a perceptron. Perceptron receives signals from each other and pulse a signal as an output of an activation function (f) . Each neuron has a bias(b), weight(w). And the activation function as shown in figure 10

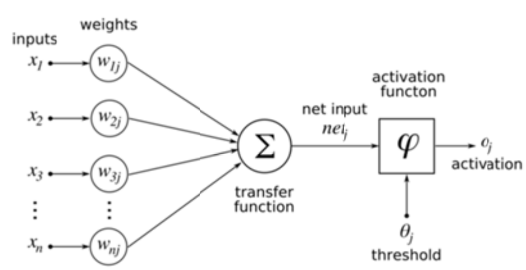


Figure 10. Single neuron

The output of the ANN is as the following function

$$y = f\left(\sum_{i=1}^{N} w_i x_i + b\right) \ldots\ldots (9)$$

The activation function gives a non – linear combination of the value between the parentheses in the above equation. Different activation functions can be used depending on the application used. Examples of some common activation functions are shown in figure 11

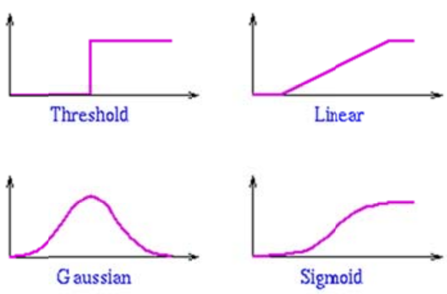


Figure 11. Some common activation functions

Source: http://staff.itee.uq.edu.au/janetw/cmc/chapters/Introduction/

a. Linearly separable case

The neural network used to solve linearly separable problems consist of an input layer with a number of neurons corresponding to the system input and an output layer with a number of neurons corresponding to the system output. This neural network creates a hyper – plane separating the two regions in the input space as shown in figure 12

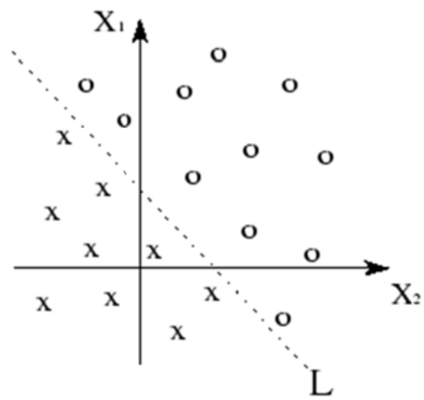


Figure 12. Linearly separable hyper-plane

Source: http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node19.html

b. Non –Linear separable case

Real – life problems are linearly non –separable, in our case we have 5 inputs for each battery (the Warburg infinite parameters and the shelf life) and one output (The energy delivered by the battery as an interpretation of its life time). Of course these types of problems can't be solved by single layer perceptron, thus we have to add hidden layers with hidden neurons. These hidden layers will produce regions separating different classes as shown figure 13. The number of hidden layers and hidden nodes per layer is an application dependent and a matter of design. However; while designing a neural network, it is important to avoid a large number of hidden layers and hidden neurons because over fitting may occur. In this case, the network will learn the peculiarities of the data which might be noisy or non – representative thus affecting the generalization of the model.

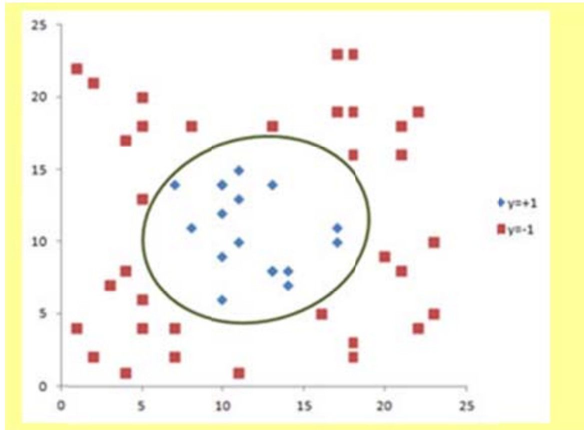


Figure13. Linearly non-separable pattern

Source: *http://people.revoledu.com/kardi/tutorial/SVM/Supervised-Learning-Illustrative-Example.html*

c. Training of neural network

Learning neural network means learning the weights and biases to best fit the input/output data. When we train the neural network, we try to find weights that can minimize the cost function. The obtained weights and the minimization are sub-optimal. Different cost functions can be used such as:

Least mean square:

$$J = \sum_{i=1}^{N} (y\, d(i) - y\, a(i))^2 \ldots\ldots\ldots \text{(10)}$$

Cross entropy:

$$J = \sum_{i=1}^{N} y\, d(i) ln ya(i) + \left(1 - yd(i)\right) \ln(1 - ya(i)) \ldots\ldots \text{(11)}$$

These cost functions compare the actual value $ya(i)$ to the desired value $yd(i)$. We solve the problem in an iterative manner. We randomly initiate the weights and the bias. Every time, the actual output is compared to the desired output to find the error. Also we will apply propagation criteria for the error. This back – propagation will modify the biases and weights using gradient descent algorithm. This process is continuously repeated until pre –defined convergence criteria are met. We will use the MATLAB ANN tool box for Warburg infinite parameters recognition and their relation with the total energy delivered till the battery ages. But before proceeding we have to make a preprocessing step.

d. Preprocessing

Before implementing any neural network, it should be trained. To train a neural network a training set should be used to help in battery aging classification. First we should normalize the data. The first attribute in the data set is just the parameters and shelf life of the batteries; hence it is not part of our training set. The other 5 features are real number values, so we have to use the standard Min Max normalization formula (the data set and the normalized values are in Appendix D)

$$B = \frac{(A - min\ A)}{(max\ A - min\ A)} \ldots\ldots (12)$$

Where B is the standardized value, A the given value. Proceeding we get all our input data in the range between 0-1. The selection of a multi-layer perceptron is for its advantageous characteristics. It is ability to model the complex functions as it is robust that it ignores the irrelevant data or noise. Another thing is that it is adaptive to environmental changes by adapting the weights and topology. Add to these advantages its easiness to use.

4. Support Vector Regression for battery parameters

To understand the Support Vector Regression, the Support Vector Machine (SVM) should be explained first. Support vector machine (SVM), originally developed by Vapnik , is based on the Vapnik-Chervonenkis (VC) theory and structural risk minimization (SRM) principle[19]. SVM on the other hand is a machine learning that is used extensively in classification problems by trying to find the tradeoff between the minimizing the optimal minimum of the training set error and at the same time maximizing the margin. It simply a supervised learning algorithm that classify the linear and nonlinear data by maximizing the margin between the support points and nonlinear mapping to transform the original data to a higher dimensions using Kernel functions. For linear data set of N training samples (x_i , y_i) Where i= 1,2,……,n, A hyper plane in the feature space can be described as $w_i x + b = 0$, where w is the orthogonal vector and b is the bias. For a linearly separable training data set, SVM generates a hyper plane that maximizes the margin between the two classes as shown below. The hyper plane is placed midway between the two classes.

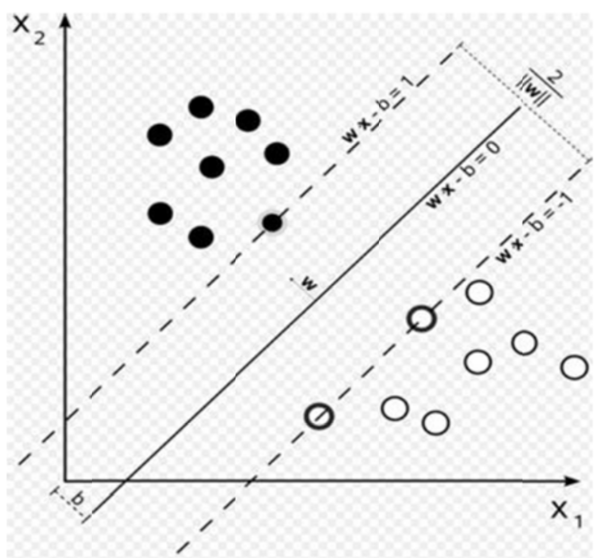


Figure 14-a. SVM hyper plane for linearly separable data

Since the minimum distance between the closest point and the hyper plane is $\frac{1}{\|w\|}$, and the margin as shown is equal to $\frac{2}{\|w\|}$, hence our target is to maximize $\frac{2}{\|w\|}$. So we can write our objective equation which is

Min $\frac{1}{2}\|w\|^2$, subject to $y_i(w_i\ x + b) \ll 1$. But if the training data are not linearly separable, then there is not a hyper-plane that could linearly classify the data. In this case when the linear SVM doesn't accomplish the job, the nonlinear SVM is used. This is done using Kernel functions which transform the feature vector to a higher Hilbert dimensional feature vector, by nonlinear mapping in which the optimal hyper plane is found. Kernel function transform x to Φ(x) by k (xi, xj) = Φ (x_i). Φ (xj). Many forms of Kernel can be used but the most common one is the Gaussian radial basis function (RBF) and the polynomial Kernel function.

Kernel Gaussian (RBF) $= \exp -\frac{\|x_i - x_j\|^2}{\sigma^2}$

Kernel polynomial function = $(x_i . x_j + 1)^p$

Here the famous Gaussian RBF Kernel function to transfer the non-separable nonlinear data to a higher dimension at which will be linearly separable will be used. The Support Vector Machine can be implemented to be applied as a regression method as shown in Figure14-b. Regarding the main characteristics that describes the maximal margin. The same main principles for the SVM are used with few distinctions. First the output of the SVR is a real number with infinite possibilities that make the prediction a hard task. A tolerance margin (epsilon) is recommended. And the algorithm is even more complicated than the SVM algorithm.

But the same concept is the same which is the target is to minimize the error by individualizing the hyper-plane that maximizes the margin, with minimum tolerated error

a. Concept of SVR

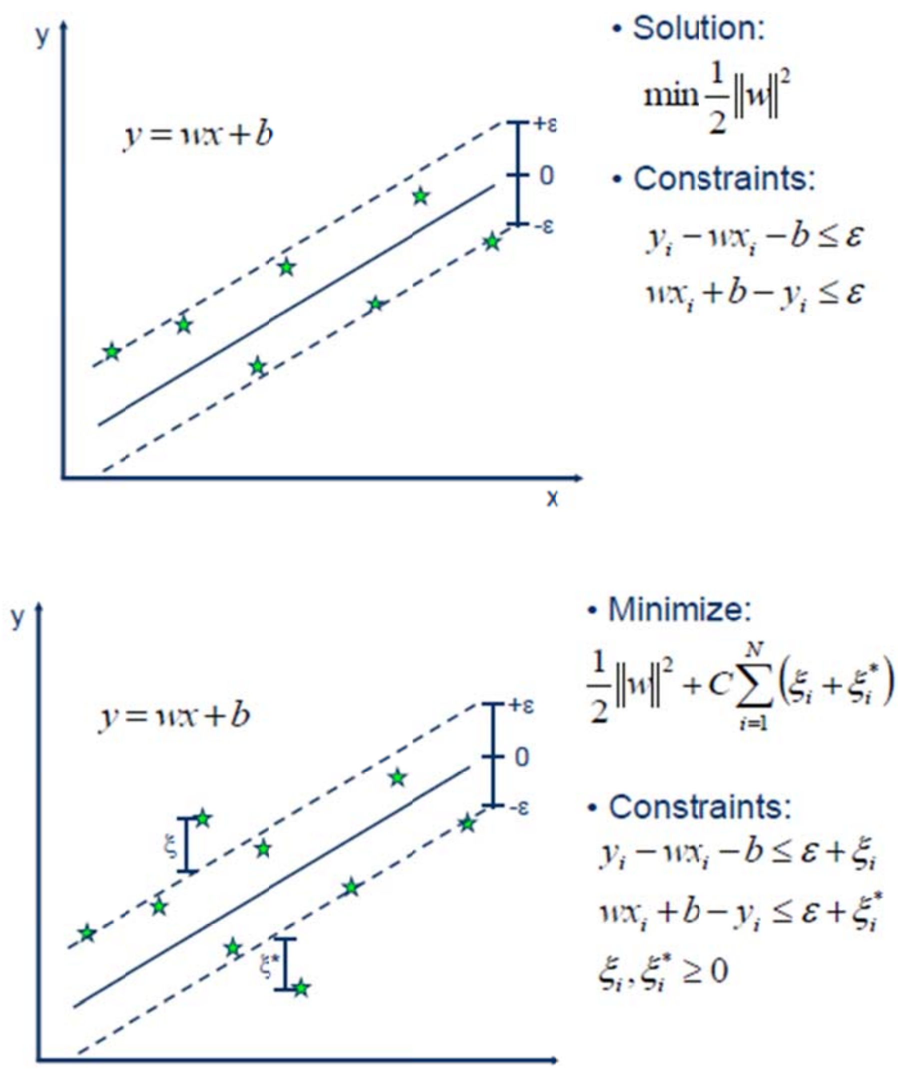


Figure 14-b. Support Vector Regression (SVR)

b. Types of SVR

i. Linear SVR

Based on the equation

$$y = \sum_{i=1}^{N}(a_i - a_i^*) \, . \, (x_i \, , x) + b \ldots\ldots (13)$$

ii. Non Linear SVR

If the data are not separable, the kernel functions transform the data to a higher dimensional space where the linear separation is possible

$$y = \sum_{i=1}^{N}(a_i - a_i^*) \,.\, (\emptyset\,(x_i)\,, \emptyset(x)) + b \text{.......} \quad (14)$$

Two types of kernel functions are available, the polynomial function and the Gaussian radial basis function.

The polynomial function and the Gaussian radial basis function as mentioned above.

In the next chapter a full experimental set –up description is going to be discussed with the experimental data measured and the results we come up with to end up with the conclusions at the end of the thesis.

CHAPTER V

EXPERIMENTAL SET–UP

Here we will describe the experiment and its set-up, and provide detailed explanation of the devices used to accomplish this experiment. Also the code used to run the devices and the algorithm that is used to extract the parameters for each model. The main purpose of the experiment is to estimate the life time of type HW, 12 V, & 7Ah lead – acid battery, by monitoring the variation of its internal parameters. As we are going to investigate the proper model that best fit the measured data.

The devices used are:

1- Programmable power supply.

2- An electronic load.

3- BRS device

4- PC

5- Temperature sensor

6- Lead-acid battery-12V& 7Ah.

A detailed explanation of the above devices will follow.

A. Description of the experiment

1. Programmable Power Supply

The dc programmable power supply (PPS), shown in Fig. 15, is a Keysight – N5767A, 60V/25A, 1500W, which is power – factor corrected and operates for a wide AC voltage range. The output voltage and current are continuously displayed on the LED indicators as they show the operating condition of the power supply. It has two panels’, front panel and rear panel. The front panel permits the user to set and control the output parameters, over-voltage, under-voltage and over-current protections ranges, and continuously monitor the settings.

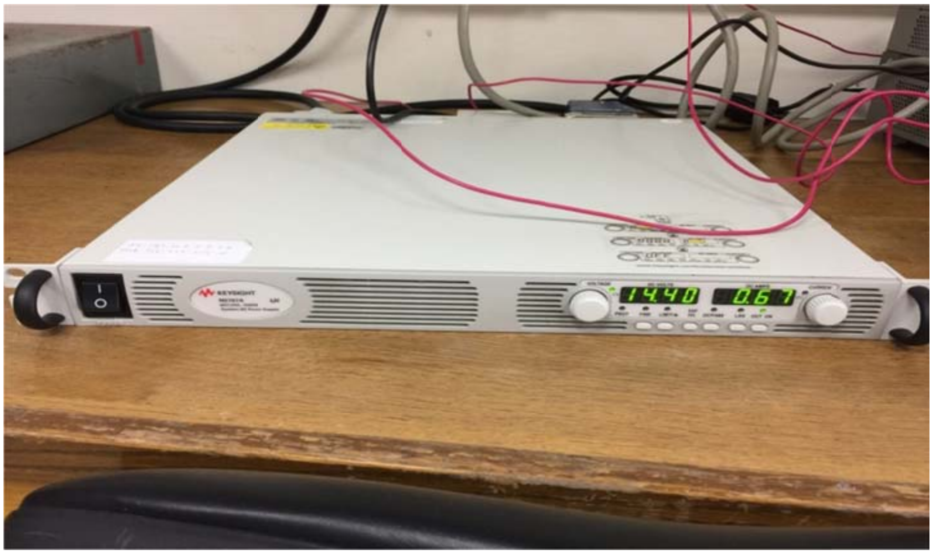


Figure 15. The programmable power supply

While the rear panel involves the connectors that control and monitor the power supply operations, by using the analog interface or the built-in remote communication interfaces (GBIP, LAN or USB). The power supply was operated in two modes, a constant voltage and a constant current mode. In the constant voltage mode, the power supply regulates the output voltage while the load current varies, however; in the constant current mode, the power supply regulate the output current at a specific value while the output voltage varies according to the load. The mode at which the power supply operates relies on some factors like the voltage setting, current limit setting and the resistance of the load. We can set the limit current or voltage limit according to our usage. Power supply can be locally controlled by turning the knob to the required settings and setting the over voltage protection, over current protection and using analog interface. The power supply also can be remotely controlled with the SCPI commands or writing the proper code that drives the (PPS). In our case we used a USB, GBIP remotely controlled interface to control the charging the battery by writing the proper code. We can control this type of power supplies by visual basic programming code, C programming code or by Lab view. In this thesis a C programming code to remotely control the whole devices was developed.

2. Electronic Load

It is a key sight N3301 A, system DC electronic load mainframe that is half rack width with the two slots only for load modules. It can dissipate up to 300 watts per slot, and for a full load mainframe the power dissipated can reach up to 600 watts. N3301A electronic load module can operate in a constant current mode (CC), constant voltage mode (CV), or constant resistance mode (CR). Further each input can be turned on or off (open -circuited), or short- circuited, see figure 16.

Like a programmable power supply, the electronic load has many features, like the different constant modes mentioned above. And has built-in GPIB interface with the SCPI command language for remote controlling, triggered input and measurement functions, independent channel operation, and overvoltage and over current and over temperature protection, and fan speed control to mitigate the acoustic noise under light load.

The front panel keyboard controls the input voltage, current and resistance. Its display supplies digital readings of input functions. Annunciators display the operating case of the electronic load. The front panel keys can access and control the electronic load function menus, they let you select and enter the values of the parameters. Like programmable power supply it can be controlled remotely by a proper programming code, like visual basic or C language code and lab view.

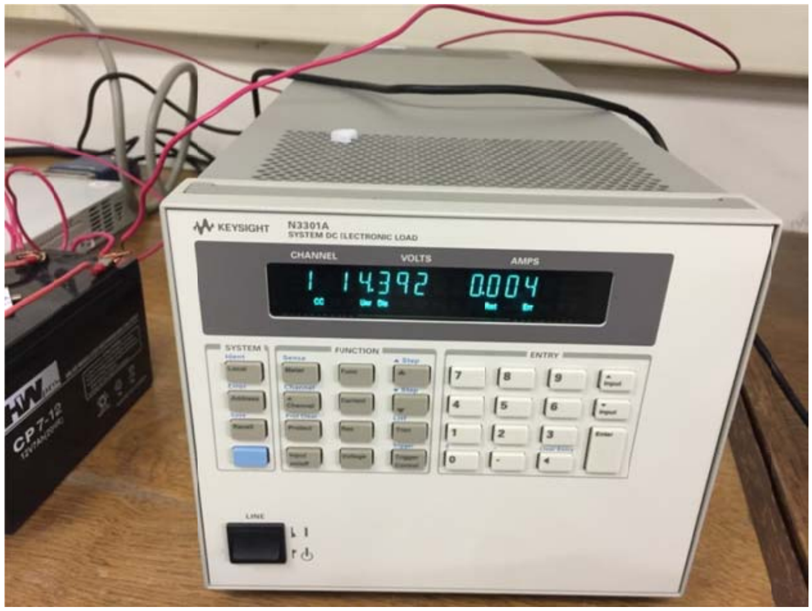


Figure 16. Electronic Load

3. Electro-impedance spectrometry (EIS)

The EIS measurements were done using the BRS device (shown in Fig. 17) that was used to supply us with a log file of the measurements of the impedance of the battery when it is needed to be measured; it gives the real and imaginary impedance of the battery as well as the magnitude and the phase, and the open circuit voltage. BRS also can give the impedance over a wide range of frequencies from 1HZ to 1000 HZ according to the frequency grid we chose, as illustrated in Fig. 18 below that shows its user interface.

Figure 17. The BRS device used for electro-impedance spectrometry

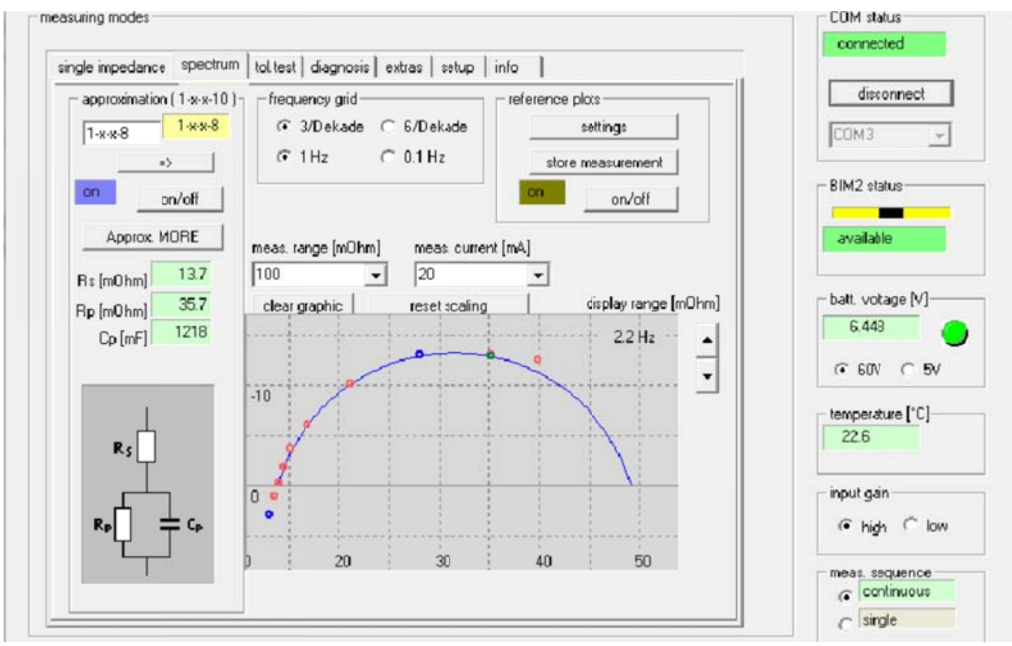


Figure 18. The Spectrum of the impedances by the BRS device

Further, it supplies the spectrum of the battery according to the model proposed. An EIS measurement is carried out when the number of predetermined cycles is done. It sends a number of signals at different frequencies into the tested battery and measures its real and imaginary impedance. These measurements are captured on a log file. Fig (19) below is the log file of measurements done by BRS. The BRS device can operate locally by connecting it directly to the battery when the cyclic process is accomplished, or we can control it remotely by writing a proper code to turn it on to be connected to the battery and provide us with the required measurements.

```
25.10.2016 11:27:08   ClearLogFile
11:50:04    1000 Hz Re:  23.290  Im:  -3.004  Be: 23.483  Ph:  -7.349  V:  13.176  T:22.7   OVL

             500 Hz Re:  26.256  Im:  -3.412  Be: 26.477  Ph:  -7.405  V:  13.174  T:22.7   OVL

             250 Hz Re:  27.970  Im:  -3.457  Be: 28.182  Ph:  -7.047  V:  13.174  T:22.7   OVL

             100 Hz Re:  30.173  Im:  -5.149  Be: 30.609  Ph:  -9.685  V:  13.172  T:22.7   OVL

            45.5 Hz Re:  32.819  Im:  -6.458  Be: 33.449  Ph: -11.132  V:  13.172  T:22.6   OVL

            21.7 Hz Re:  37.205  Im: -10.035  Be: 38.535  Ph: -15.095  V:  13.170  T:22.7  OVL

            10.0 Hz Re:  47.654  Im: -13.531  Be: 49.537  Ph: -15.851  V:  13.170  T:22.7   OVL

             4.7 Hz Re:  57.829  Im: -13.241  Be: 59.325  Ph: -12.896  V:  13.170  T:22.7   OVL

             2.2 Hz Re:  65.868  Im: -11.752  Be: 66.908  Ph: -10.116  V:  13.167  T:22.6   OVL

             1.0 Hz Re:  73.496  Im: -12.176  Be: 74.498  Ph:  -9.406  V:  13.167  T:22.7   OVL<<==
```

Figure 19. The log file supplied by the BRS device

A personal computer with a screen is needed for the display of the instant voltage and current during charging and discharging process. The code written requires a display of the voltage and the current of the battery every single minute throughout the experiment

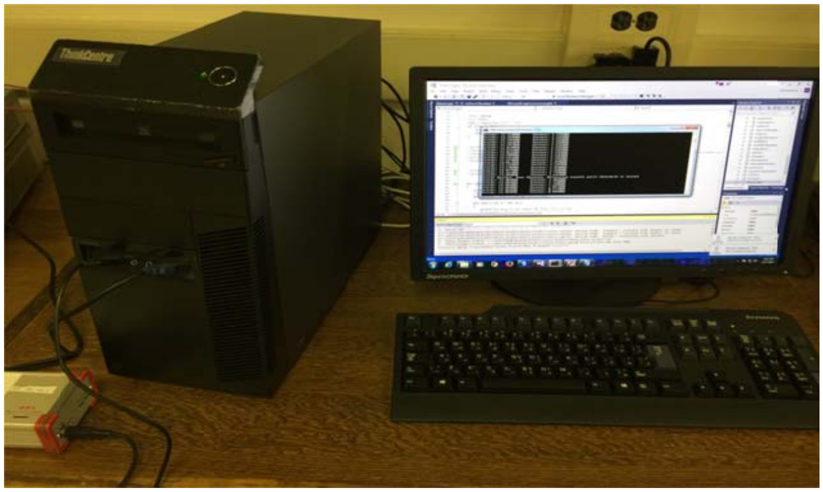


Figure 20. The PC

4. Temperature Sensor

The temperature sensor shown in figure 21 is a small sensitive device that can be attached to the battery or even put on the table holding the devices to measure the ambient temperature throughout the experiment since we are performing the cyclic method under a constant temperature for about 22.7°C $\pm$ 0.3°C.

Figure 21. The Temperature Sensor

5. Lead –acid battery

It is a HW 12 V, 7Ah lead acid battery, shown in figure 22, which uses the constant current or constant voltage (CC/CV) mode. A constant current is used to raise the terminal voltage to its upper voltage limit, at that point the current starts to drop due to saturation. The charge time is about 12-16 hours. Lead acid batteries have three stage charging process. They are:

1- Constant current charge: This applies the bulk of charge, about 70% of charge and consumes half of the required charge time (about 5-8 hours).
2- Topping charge: it continuous at a lower charge current and constant voltage for the remaining 30% charge to reach saturation and it takes about 7 -10 hours.
3- Floating charge: that compensates for the self- discharge loss to maintain the full charge status of the battery.

Figure 23 shows the three stages to fully charge the lead-acid battery

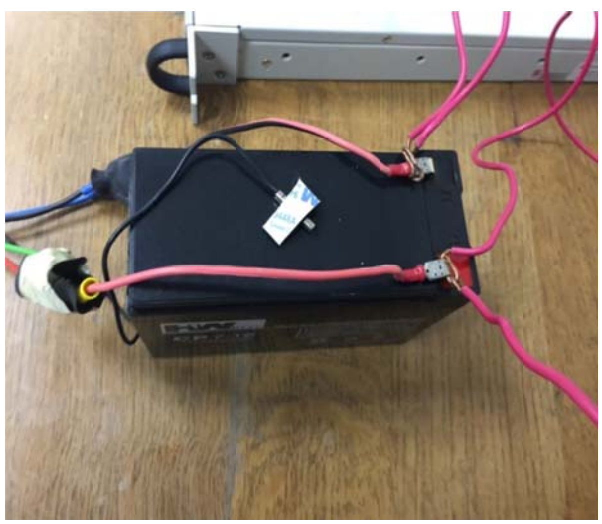


Figure 22. The lead - acid battery

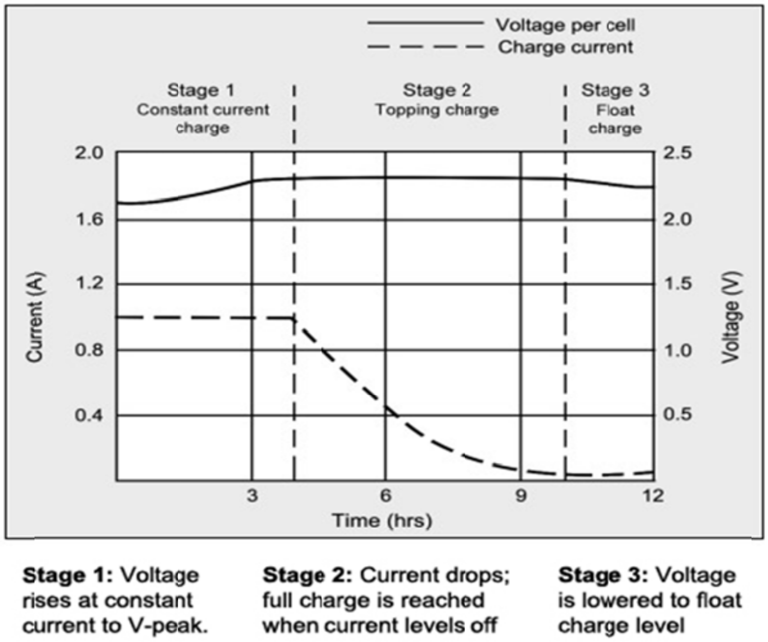


Figure 23. The charging stages for lead -acid battery

B. Experimental procedure and methodology

The experiment set-up is illustrated below in Fig. 24. The programmable power supply is connected by GPIB with the electronic load and USB to the PC; hence the programmable power supply and the electronic load are now connected to the PC. Also a parallel connection is done between these two devices and the lead- acid battery under test. EIS is connected also to the battery via a connection wire (Kelvin clips) and to the PC by a USB. After that, the temperature sensor is attached to the battery

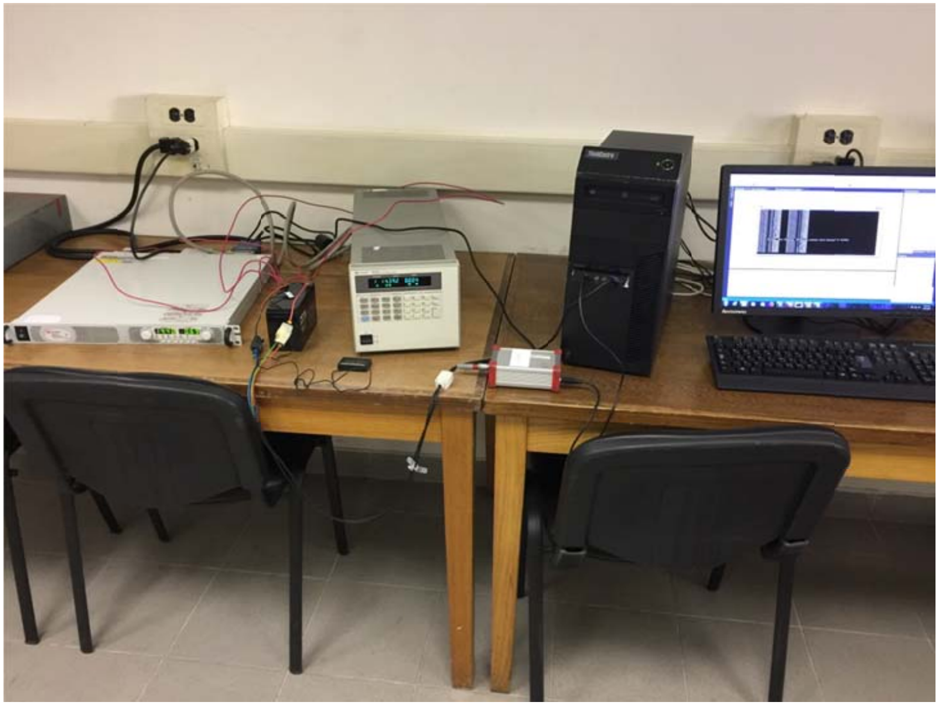


Figure 24. The experimental set up

The process to be followed is as follows:

1- The electronic load and the BRS should be turned off.

2- We then start to charge the new lead –acid battery at a constant current equals I_4 = 7/4 = 1.75 A using the programmable power supply until the voltage display on the LED front panel on the programmable power supply reach 14.4V.

3- The bulk stage is followed by a topping charge at a constant voltage of 14.4 V and the current starts to decline until it reaches I_{40} = 0.175A.

4- In our experiment we will not consider the floating charge stage for the sake of speeding the experiment since we repeated the experiment for three different samples.

5- After finishing the charging process the power supply will be turned off, and let the battery rest for 15 minutes.

6- The BRS will turn on to take the readings of the real and imaginary impedance at different frequencies at ranges from 1HZ to 1000 HZ for the new battery.

7- Then the battery cycling begins.

8- We repeat step 2 and 3 then the power supply turned off and the electronic load turned on to start the discharge process at I_4 = 1.75 A until the voltage reaches 1.75 V/cell i.e. 10.5 V (DOD = 100%). Current and voltage are continuously monitored at 1 minute intervals.

9- Step 8 is repeated 20 times.

10- Then the battery is charged as in steps 2 &3 and the battery is left to rest for 15 minutes and after that the impedance readings are taken using the BRS device.

11- Steps 7, 8, and 9 are considered to be one loop.

12- And then we repeat the loop 10 times. Till we reach number of cycles = 20*10 =200 cycles.

13- According to the data sheet, the Lead acid battery after 200 cycles has aged where in each cycle the depth of discharge was 100%.

14- We wrote a C programming code to control the charging – discharging process and recording data from BRS.

15- Since the charging and discharging are time varying process according to the battery status, some data should be taken at different times, midnight or at dawn. So, team viewer software was installed and the devices were controlled remotely by a team viewer program, also we can constantly check the cyclic operation at any time and any place.

16- After finishing the 200 cycles we select the model to be tried and fit the data and use to process them using the MATLAB algorithm early developed. The parameters of the model are obtained using least square method using the function “fminunc” to minimize the error between the measured data and the model.

C. The code used to remotely control the experiment

It is a C code that perform the steps mentioned above (for programmable power supply, electronic load and BRS) is attached in the appendix (A)

D. Parameter estimation

To find the parameters of the battery, after applying the cyclic method and take the measurements of Z_R andZ_{Im}. Suppose that we choose the single Randle cell to model the battery. Then as previously mentioned the equivalent impedance Z_{eq} is given by:

$$Z_{eq} = R_s + \frac{R_p(\frac{1}{j\omega C_p})}{R_p + \frac{1}{j\omega C_p}} = R_s + \frac{R_p}{1 + j\omega C_p R_p}$$

$$= R_s + \frac{R_p}{1 + j\omega C_p R_p} \frac{1 - j\omega R_p C_p}{1 - j\omega R_p C_p}$$

Now by let $R_p C_p = \tau_p$ the expression can be written as:

$$Z_{eq} = R_s + \frac{R_p(1 - j\omega\tau_p)}{1 + \omega^2\tau_p^2}$$

With real and imaginary components given by:

$$Z_R = R_s + \frac{R_p}{1 + \omega^2 \tau_p^2} \qquad (15)$$

$$Z_{Im} = \frac{-\omega R_p \tau_p}{1 + \omega^2\tau_p^2} \qquad (16)$$

So we will cross-multiply the above two functions to avoid differentiating fractional expressions. In this case (15) and (16) will yield the following, respectively:

$$Z_R\left(1+\ \omega^2\ \tau_p^2\right)=\ R_s\left(1+\ \omega^2\ \tau_p^2\right)+\ R_p$$

$$Z_R\ =\ R_s\left(1+\ \omega^2\ \tau_p^2\right)+R_p-\ Z_R\,\omega^2\ \tau_p^2 \quad (17)$$

$$Z_{Im}\ (1+\ \omega^2\ \tau_p^2)=\ -\omega R_p\ \tau_p$$

$$Z_{Im}=\ -\omega R_p\ \tau_p-\ Z_{Im}\omega^2\ \tau_p^2 \quad (18)$$

Each one of (17) and (18) represents an element of a vector of measurements corresponding to a different frequency. When written in matrix form they appear as follows:

$$\underline{Z}=\underline{h}\left(\underline{x}\right)\ +\ \underline{e}$$

Where:

$$\underline{x}=\begin{bmatrix}R_s\\R_p\\\tau_p\end{bmatrix},\qquad \underline{Z}=\begin{bmatrix}Z_R\\Z_{Im}\end{bmatrix},\quad \underline{h}\left(\underline{x}\right)=\left[\frac{\underline{h}_1\left(\underline{x}\right)}{\underline{h}_2\left(\underline{x}\right)}\right]$$

Our objective is to determine the vector $\underline{x}$ to minimize the sum of square of errors denoted by $J=\underline{e}^t\ \underline{e}$ which can be written as follows:

$$J=(\underline{Z}-\ \underline{h}(\underline{x}\ ))'(\underline{Z}-\underline{h}\ (\ \underline{x}\ )) \quad (19)$$

The optimum value $\underline{x}^*$is obtained by solving by solving the system of partial derivatives of J with respect to the state variable $\underline{x}$ equated to the null vector:

$$\left.\frac{\partial J}{\partial\ \underline{x}}\right|_{x=\ x^*}=-\,2\ \underline{H}'\ \left(\ \underline{x}^*\right)(\ \underline{Z}-\ \underline{h}\ (\underline{x}^*\ )=0 \quad (20)$$

Given a point x_0 close to the solution, then;

$\underline{h}\ \left(\underline{x}\right)$ is approximated by Taylor series

$$h(\underline{x}^*)=h\left(\underline{x}_0\right)+\ \underline{H}\left(\underline{x}\right)\Delta\underline{x} \quad (21)$$

where

$$\Delta\underline{x}=(\underline{x}^*-\ \underline{x}_0)$$

$$\underline{H} = \frac{\partial \underline{h}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial R_s} & \frac{\partial h_1}{\partial R_p} & \frac{\partial h_1}{\partial \tau_p} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \frac{\partial h_2}{\partial R_s} & \frac{\partial h_2}{\partial R_p} & \frac{\partial h_2}{\partial \tau_p} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Now the elements of the $\underline{H}$ matrix are given by:

$$\frac{\partial h_1}{\partial R_s} = (1 + \omega^2 \tau_p^2)$$

$$\frac{\partial h_1}{\partial R_p} = 1$$

$$\frac{\partial h_1}{\partial \tau_p} = 2\omega^2 R_s \tau_p - 2Z_R \omega^2 \tau_p$$

$$\frac{\partial h_2}{\partial R_s} = 0$$

$$\frac{\partial h_2}{\partial R_p} = -\omega \tau_p$$

$$\frac{\partial h_2}{\partial \tau_p} = -\omega R_p - 2Z_m \omega^2 \tau_p$$

From (20) and (21) we obtain after expansion:

$$\underline{H}'(\underline{x}^*)\, \underline{Z} - \underline{H}'(\underline{x}^*)\, \underline{h}(\underline{x}_0) - \underline{H}'(\underline{x}^*)\, \underline{H}(\underline{x}^*) \Delta \underline{x} = 0$$

Let $\underline{Z} - \underline{h}\left(\underline{x}_0\right) = \Delta \underline{Z}$ then

$$\underline{H}'(\underline{x}^*)\, \Delta \underline{Z} = \underline{H}'(\underline{x}^*)\, \underline{H}(\underline{x}^*)\, (\underline{x}^* - \underline{x}_0)$$

By rearranging we obtain:

$$\underline{x}^* = \underline{x}_0 + [\underline{H}'(\underline{x}^*) \underline{H}(\underline{x}^*)]^{-1}\, \underline{H}'(\underline{x}^*) \Delta \underline{Z}$$

So if x_0 is not very close to the solution we may need to iterate as follows:

$$\underline{x}_{r-1} = \underline{x}_r + [\underline{H}'(\underline{x}_r)\, \underline{H}(\underline{x}_r)]^{-1}\, \underline{H}'(\underline{x}_r)\, \Delta \underline{Z}$$

Alternatively we may use the MATLAB function "fminunc" to find a local minimum of J as defined by (19) which is a function of the state variable. X = fminunc (FUN, X0) is a MATLAB function that finds a (local) minimum of a (nonlinear) multivariable function. The user must supply a function J which returns the value and partial derivatives (if the option "GradObj" is on) with respect to all variables. It starts at X0 and attempts to find a local minimizer X of the function$J$. FUN is a function that we will have to define to calculate a value of J as given by (19), which accepts input X and returns a scalar value J evaluated at X, and X0 is a vector representing the initial solution. Use the (GradObj) since the main benefit of providing gradient is that the convergence will be quicker. Anyway even without gradient, we should be able to converge to the solution within the desired accuracy but on the expense of more iteration. We extended the analysis to Double Randle cell, Warburg infinite impedance and Warburg finite impedance as well. Also we will write a code to find the parameters of Shepherd model and try to plot the discharge voltage versus time curve to calculate the total energy delivered by the battery before being disposable.

CHAPTER VI

RESULTS AND CONCLUSIONS

As mentioned in the previous chapter, many models are applied to model the battery and its life time. Cyclic method (charging –discharging process on three samples) is used. The batteries type is lead-acid, HW, 7Ah, 12 volt. The process is done at 100% depth of discharge (DOD). Here in this chapter the results of modeling the 3 batteries using Single Randle Cell, Double Randle Cell, Warburg impedance and Shepherd equation models are presented each with its parameters to decide which best fit the data measured experimentally and at the same time test the parameters for their behavior. Also monitoring and calculating the total time of discharge for each sample and calculate the total energy delivered by each battery by energy integration method.

A. Life time assessment by studying the total discharge time of the battery

We implemented the data acquired from the experiment which is the voltage, current and time since a display of the output voltage and current every single moment was recorded. The data of the time of discharge each loop (each loop corresponds to 20 cycles of charging and discharging) is plotted versus the number of cycle's i.e. a plot of the total time required for charging- discharging process every 20 cycles. Doing so, until the battery ages at 200 cycles.

For example, Figure 25 presents the amount of time that each battery take to discharge over 20 cycles after they have completed the given number of cycles (x-axis). From figure (24) we can see that the total discharge time for the first battery when the battery aged (200 cycles as mentioned in the data sheet of the Lead-acid battery) is 7.8

hours, which is approximately equal to 23 minutes, the time it takes the voltage of the battery to drop to 1.75 V/cell that is equivalent to 10.5 V each single cycle. The same can be generalized for the second battery. However the third battery takes a total discharge time when aged (200 cycles) equals 12.5 hrs. That is on average about 37.5 minutes in each cycle for the voltage to decline to 10.5 V. To confirm our observations, we tried to investigate the plot volt versus time discharge curve.

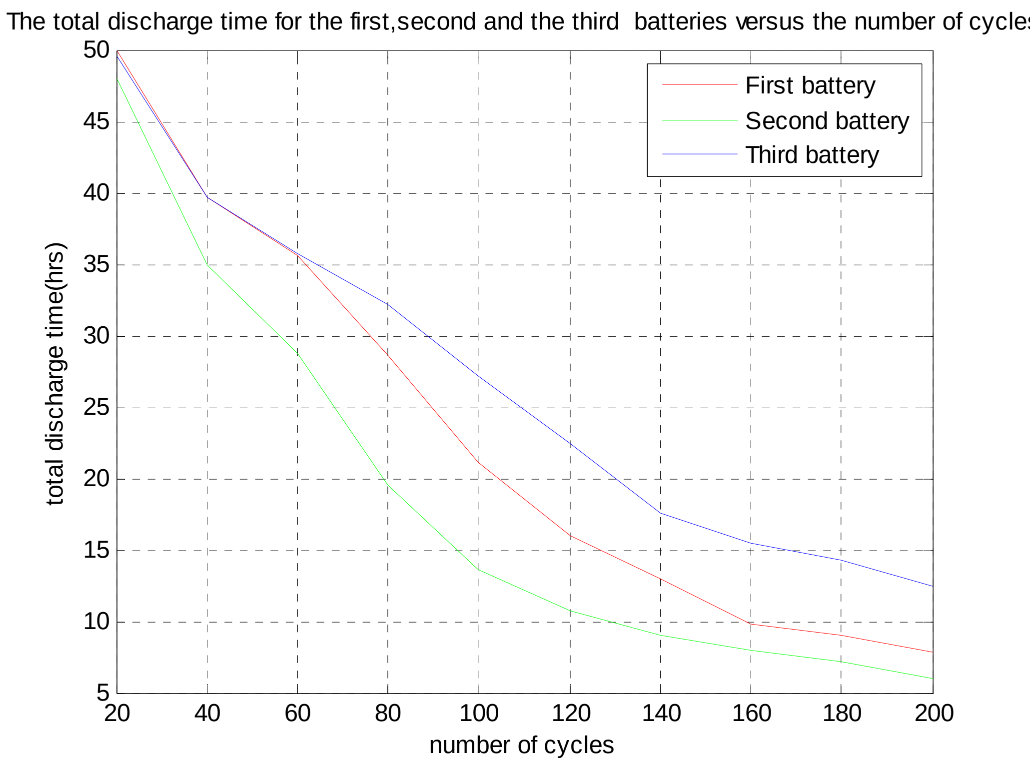


Figure 25 the total discharge time for the three batteries throughout the whole experiment

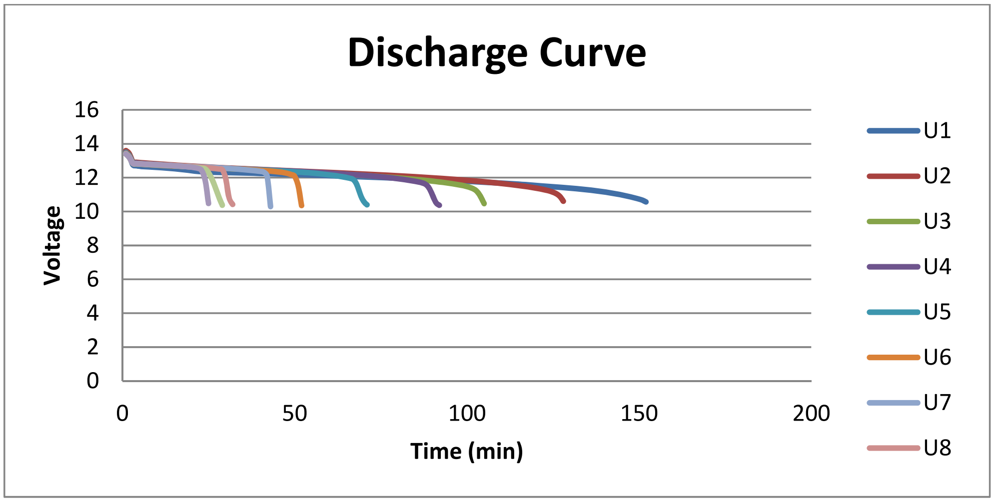


Figure 26. The discharge curve for the first battery

It is obvious that the time it takes the battery voltage to drop to 10.5 V is 20 minutes for a full battery discharge when it has aged and need to be replaced. The tests on the two other batteries confirm the information about the total discharge time. The reason that the third battery takes a longer time for the voltage to drop to 10.5 V is that this battery was newly manufactured and hasn't been shelf-stored like the first and second one, and so was not as aged through self-discharge. Here it is worth noting that according to the lead acid battery data sheet, the battery's capacity decline by 3% monthly at 20°C due to self-discharge.

So a first method to determine whether the battery is to be replaced or not is to discharge it at I_4 and monitor the output voltage across its terminals, if the voltage drops to below 10.5V in a time less than 20 minutes, we can conclude that the battery has fully aged and should be replaced, otherwise it is still usable.

B. Life time assessment by Energy Integration method

As explained in Chapter (3), this process relies on the value of the integral obtained from the V- t discharge curve. Recall figure 26, we calculated the average energy delivered each loop (one loop corresponds to 20 cycles of charging and discharging) and hence for the total 200 cycles. This is done for the three batteries to come up with the following:

The first battery delivered about 4440 KWh until fully aged, while the second battery delivered 3092 KWh, however the third battery transmitted 5243 KWh, this is logical since we claimed that the third battery was not shelf-stored and as such had more life-time and was able to deliver more energy throughout the experiment. Maybe we can determine the shelf life of the battery and then consider it as a correction factor for determining the life time of the battery using energy integration approach. These values were trained as an output using the Artificial Neural Network. They were used, since we can consider the percentage of the energy delivered at any time relative to the total energy delivered by the battery as an interpretation of the life time of the battery to estimate its life time.

C. Life time assessment by studying the variation of the internal parameters after battery modeling

First the single Randle cell (SRC) and the Double Randle cell (DRC) were investigated, they gave a poor modeling of the measured data, this is shown below in Fig. 27 &28.

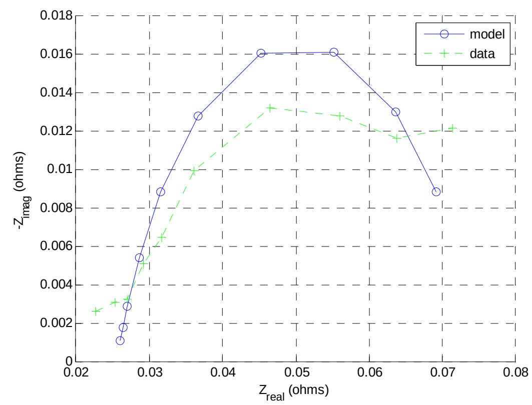


Figure 27. SRC for the first battery (140 cycles)

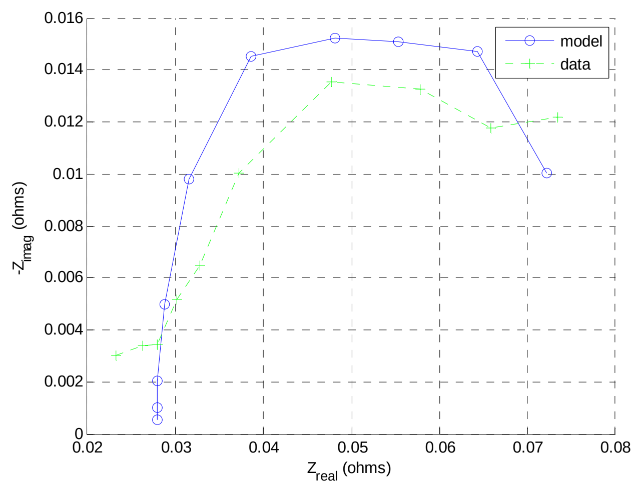


Figure 28. DRC for the first battery at 140 cycles

We can see clearly the deviation between the experimental data recorded by the BRS and the Randle models. From investigations on the SRC and the DRC parameters in the first battery, we can clearly see some erratic behavior in the values of R_s, R_p and C_p , for Single Randle Cell and a more irregular trends of R_s , R_{p1} ,R_{p2} ,C_{p1} , and C_{p2} of the Double Randle Cell model. One example on such erratic behavior is shown in the value of Fig. 29. A similar erratic behavior was observed in the values of the parallel capacitances of the DRC model in Fig.30; the two other batteries have the same outcome.

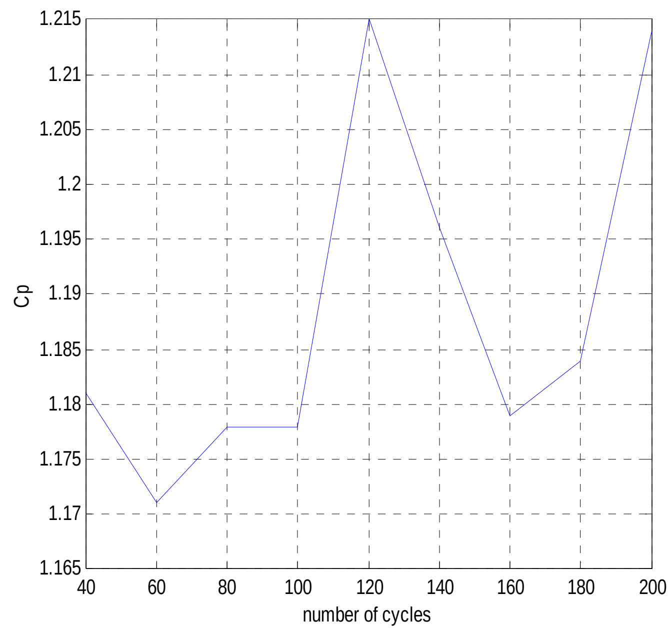


Figure 29. C_p Versus number of cycles in the first battery SRC model

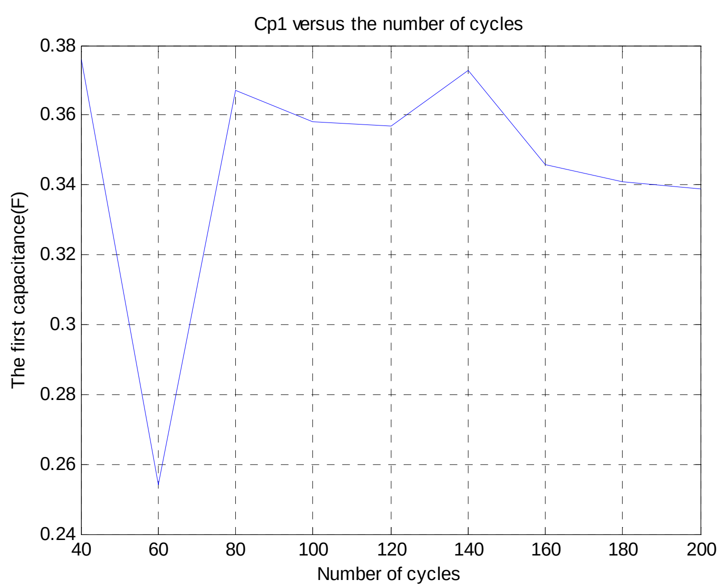


Figure 30.The erratic behavior of the capacitance C_{p1} in the DRC

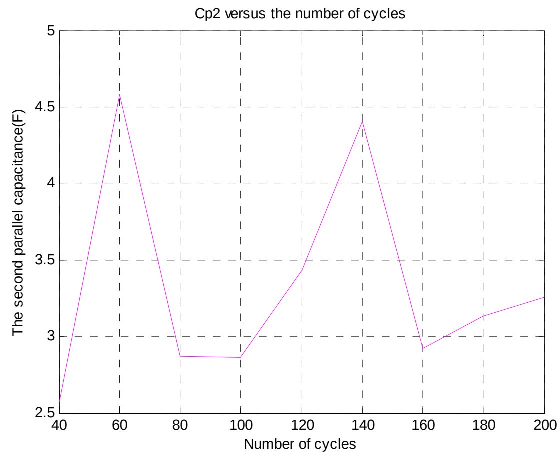


Figure 31. The erratic behavior of the capacitance C_{p2} in the DRC

We concluded from above that The Single Randle Cell and so the Double Randle Cell, cannot properly model the lead – acid battery, they poorly fit their measured data and the parameters of SRC and DRC did not show clear trends in their values as the battery aged through cycling.

Then the modified Randle cell by adding the diffusion impedance which is the Warburg impedance model is tried. The infinite Warburg impedance came up with acceptable results; take the second battery for example, we can see in figure 32, the deviation between the measured data and the Warburg impedance model. When trying to investigate the model parameters, the Warburg parameters shows undesirable trends over the frequency range from 1HZ to 1000 HZ as presented in figure 33. However; the series resistance shows an increasing trend as the number of cycles of charging and discharging increase. The parallel resistance on the other hand declines as the battery ages as obvious in figure 34. Figure 35 shows the parallel capacitance in the Warburg infinite impedance model in the second battery but with an erratic trend at cycle 120.

And so, figure 36 shows a trend in the Warburg constant impedance A_w but again with an ill-mannered behavior at cycle 120.

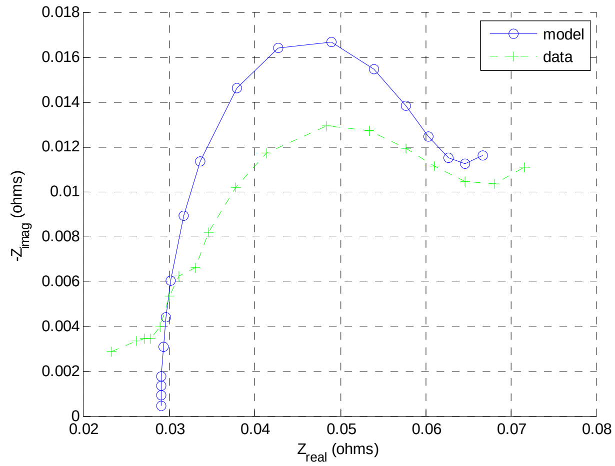


Figure 32. The clear deviation between the data and the model in Warburg infinite model

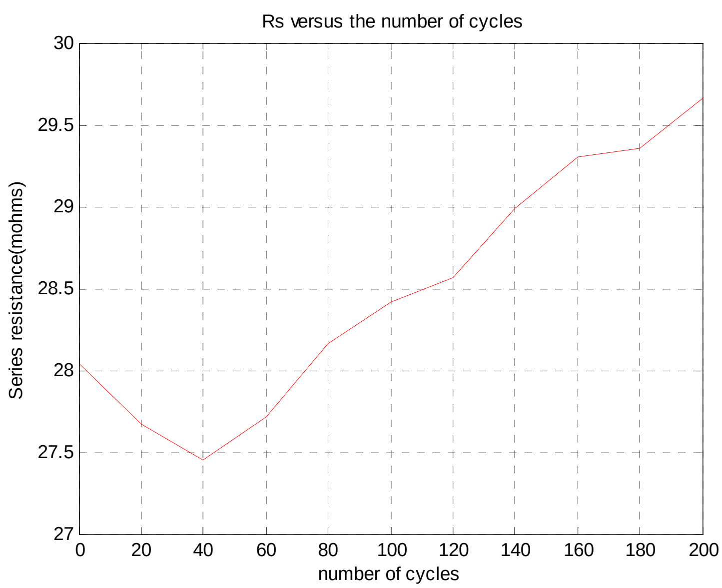


Figure 33. The series resistance of Warburg infinite impedance (second battery)

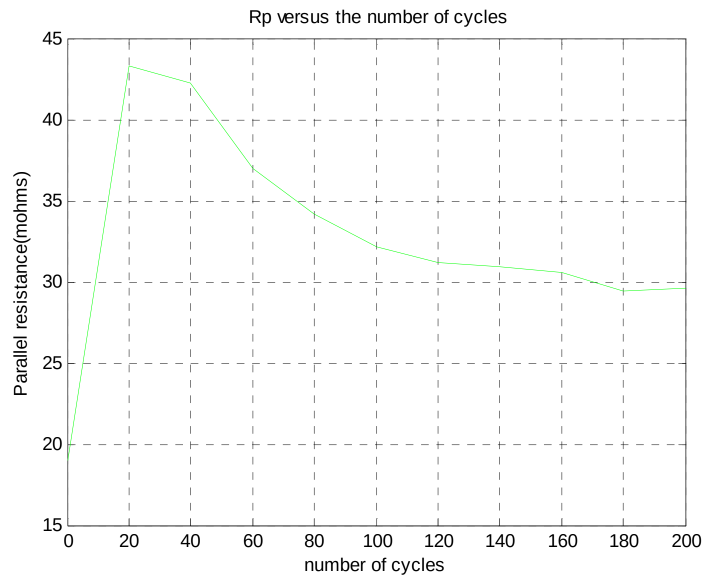


Figure 34. The parallel resistance decreases as the battery ages (second battery)

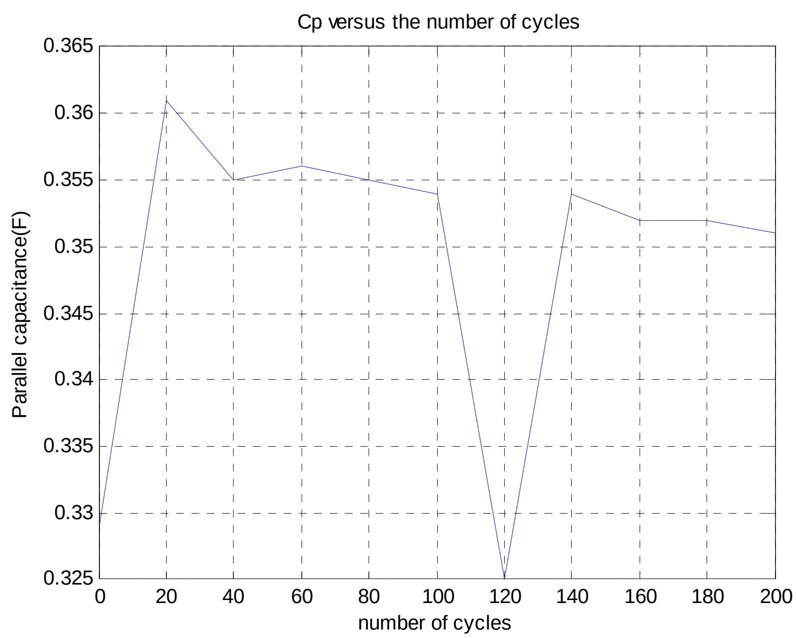


Figure 35. The irregular behavior of the capacitance as the battery ages

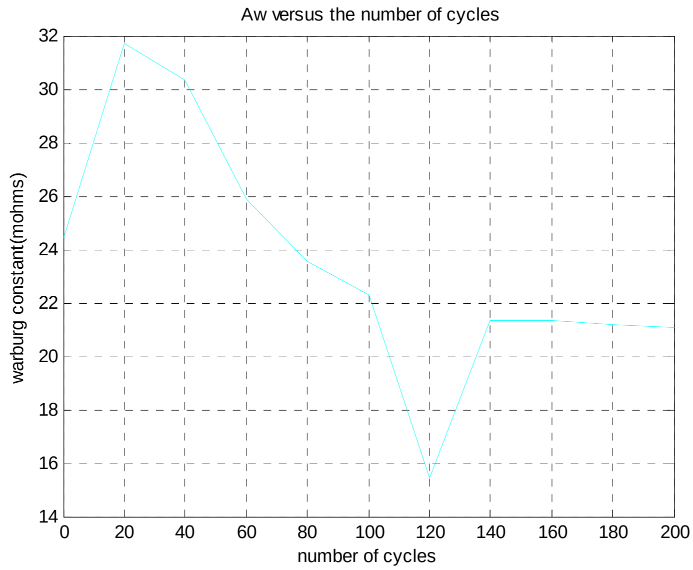


Figure 36. Aw Versus the number of cycles

We applied the finite Warburg model by adding another parameter B_w as mentioned in the previous chapter and found out that the situation didn't show any improvements. Still the measured data suffer the mismatch with the Warburg finite model, for example for the third battery at 140 cycles the relation between the real and imaginary impedances is as illustrated in figure 37. Figure 38 shows that the relation between the phase angle and the angular frequency still deviated although the magnitude perfectly matched with the model as the angler frequency increases.

The parameters of the Warburg finite model showed erratic pattern that we can strongly conclude that the Warburg finite impedance model didn't fit our data. In figure 39, the series resistance shows an ascending manner, while figure 40 demonstrates the descending behavior of R_p as the cyclic method goes on, with an ill-mannered value at cycle 40. Figure 41 shows the irregular manner of the parallel capacity; however the Warburg first constant A_w shows a descending trend with a sudden erratic behavior at

120 cycles as in figure 42. Warburg second constant B_w behavior shows nothing about the battery life time; this is obvious in figure 43

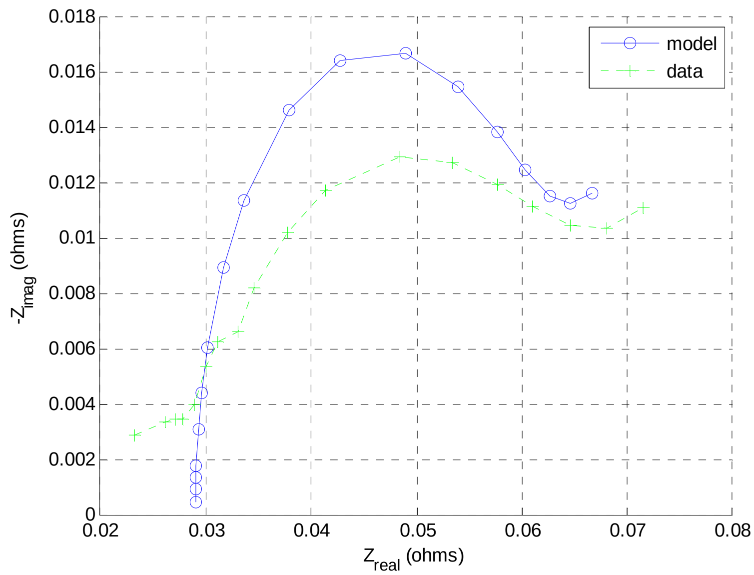


Figure 37. Z_{real} Versus Z_{imag} for the third battery at 140 cycles

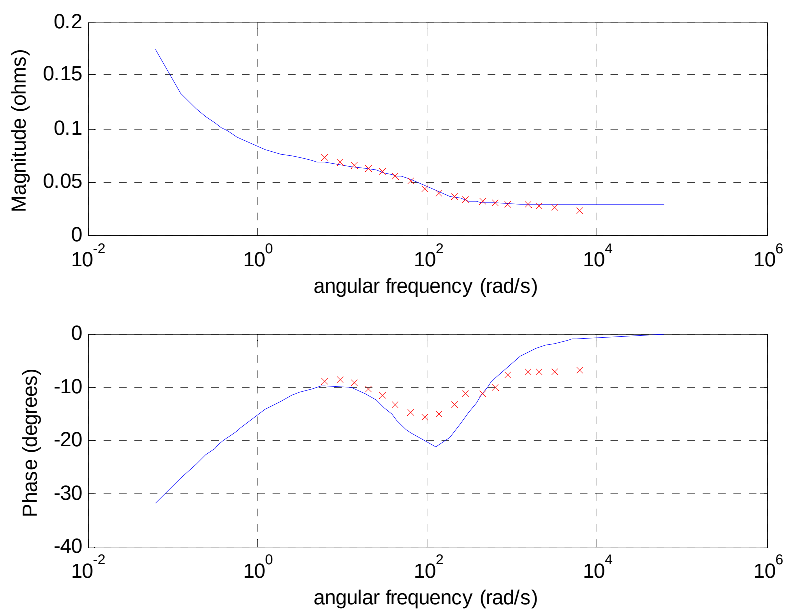


Figure 38. The magnitude and phase angles versus angular frequency

.

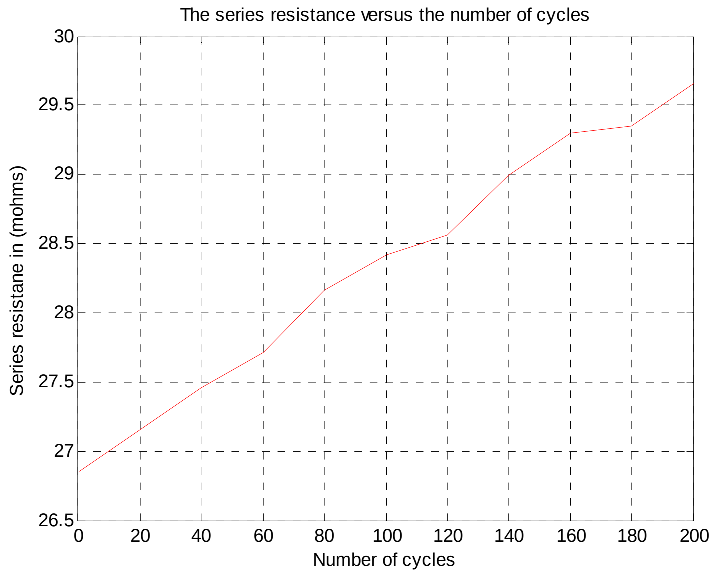


Figure 39. R_s Versus the number of cycles

Figure 40. R_p Versus number of cycles

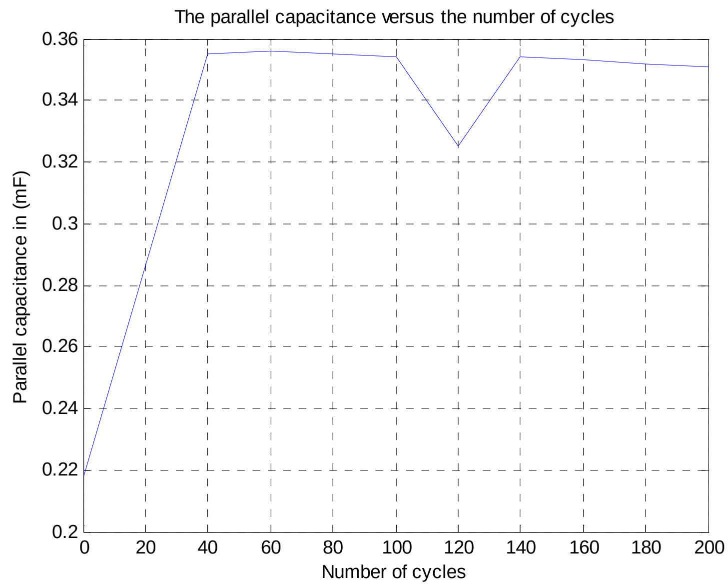


Figure 41. The irregular manner of the parallel capacitance

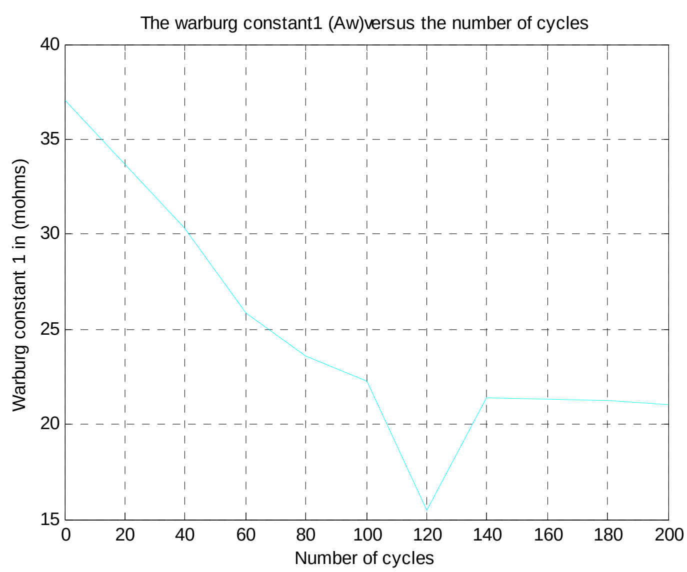


Figure 42. Warburg first constant versus number of cycles

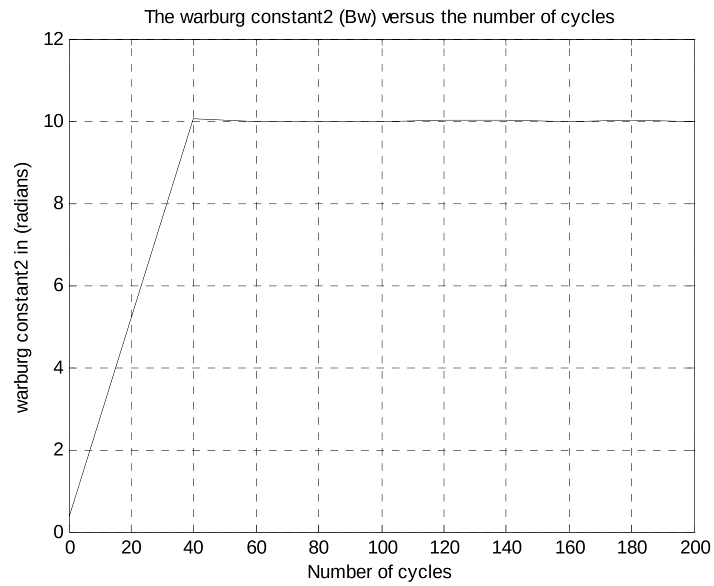


Figure 43. Warburg second constant (B_w) versus the number of cycles

The only explanation of the unfitting the previous models and the inconvenient behaviors of the parameters unlike the expectations, is that when modeling the data, filtration takes place and this affects the manners of the parameters and varies their trend. For this reason we should be aware of data filtration while modeling them.

In fact this leads us to promote our study to use the Shepherd model since this model depends on relating the open circuit voltage to the depth of discharge by the following equation (mentioned in the previous chapters)

$$U_{cell} = U_{od} - g_d(H) + \rho_d \frac{I}{C_N} + \rho_d \mathrm{M_d} \frac{I}{C_N} \frac{H}{C_{d-H}} \ldots\ldots (6)$$

This equation clearly relates the open circuit voltage with the cell voltage and the normalized state of charge H that is relied on the time of discharge. When trying to model the data using the above equation we got the relation of the measured data and the model data in figure 44

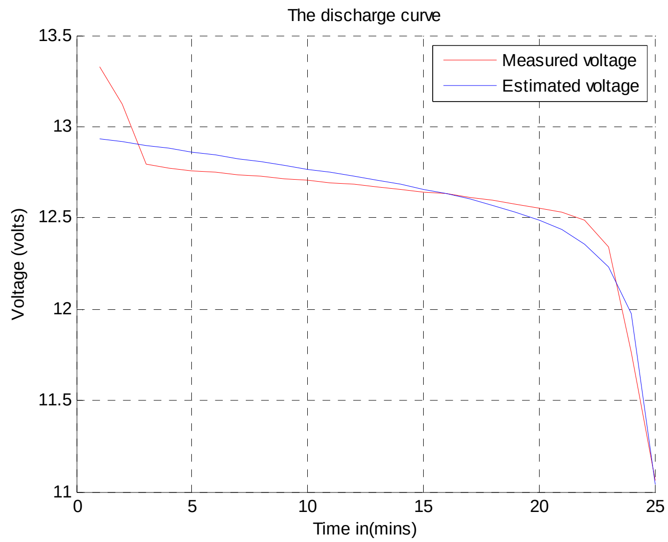


Figure 44. Shepherd model for the discharge curve at 140 cycles

We noted that the shepherd equation lacks the exponential region which is the first zone preceding the normalized one, so we have to add an exponential expression that presents the exponential part. In this case modified Shepherd equation is going to substitute the Shepherd by adding the term ”Ae^{-BH} “, where A is the difference between the voltage cell at the exponential region and the normalized region, and B can be found as the corresponding value to state of charge F=1-H at the end of the exponential zone. Introducing the exponential term will modify the trend as shown in figure 45. Regardless of the stray manner of the parameters, modified Shepherd equation shows a perfect fit to the measured (voltage) data.

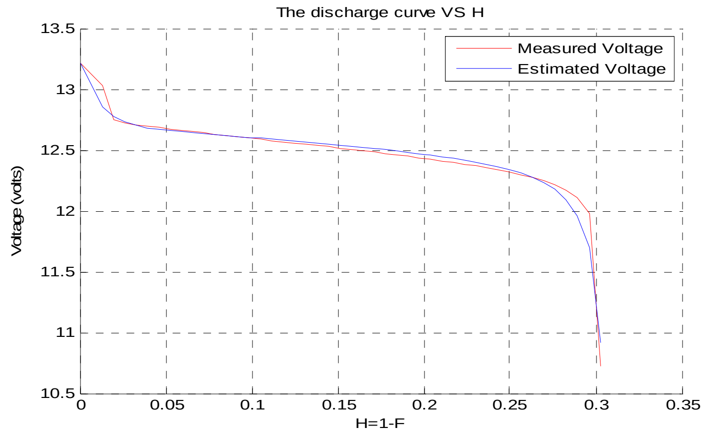


Figure 45. The modified Shepherd model

A closer look to the above models and their parameters leads us to the fact that the infinite Warburg impedance model is the closest one to model the parameters of the lead acid battery but if we narrowed the range of frequencies over which the real and imaginary impedance data are taken from 1 to 1000 HZ to be from 1 to 100 HZ, A better modeling will come up with a little erratic behavior in the parameters. This little erratic trend may occur because of noise or distortion occurred in the signals that were sent by the BRS device with different frequencies to measure the battery impedances. A more involved investigation over the range from 1 to 100 HZ to exclude the frequencies that may have caused the erratic behavior was carried out. Doing so leads to a better fit with a clearer trend pattern of the parameters of the batteries as they are cycled throughout the test.

For example, examining the second battery before excluding the frequencies that cause noise or distortion and after excluding those frequencies that is responsible for the ill-mannered trend. An investigation concerning the double layer capacitance revealed that C_p is suffering the ill- mannered trend in loop 6 (120 cycles) as well as loop2

(40cycles), examining the frequencies is carried out one by one to study the effect of excluding them one each step, we come up with the fact that 3.3 Hz in loop(6) and 4.7Hz in loop(2) causes this erratic trend Figure 46 shows the erratic behavior of the double layer capacitor before excluding those frequencies, while figure 47 shows the clear increasing trend after excluding those frequencies. The Warburg infinite constant A_w at cycle 120 also has the erratic trend but after excluding the 3.3 Hz in loop (6) and 4.7 Hz in loop (2), an improvement is clear as shown in figure 49 with a decline tendency. The parallel resistance $\boldsymbol{R_p}$ didn't suffer the ill-mannered behavior as C_p or A_w figure 51. The same can be said about $\boldsymbol{R_s}$ that has an ascending trend as in figure 53.

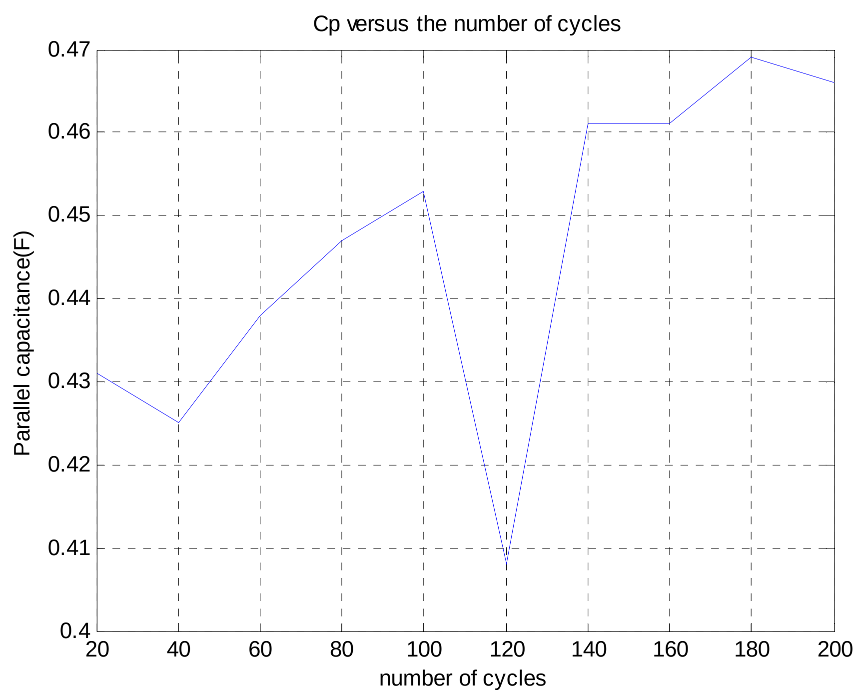


Figure 46. C_p Without excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

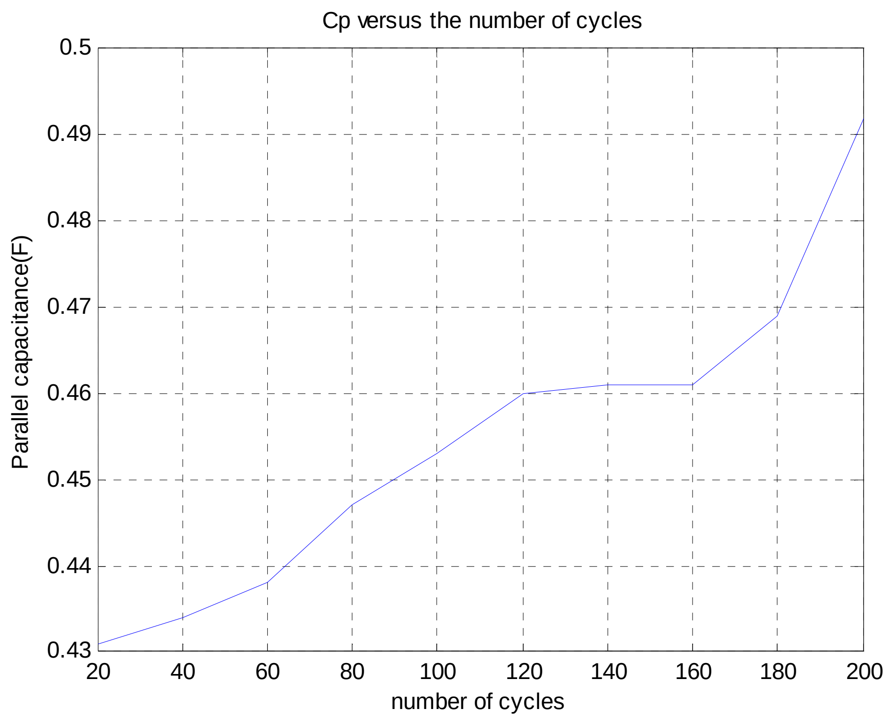


Figure 47. C_p With excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

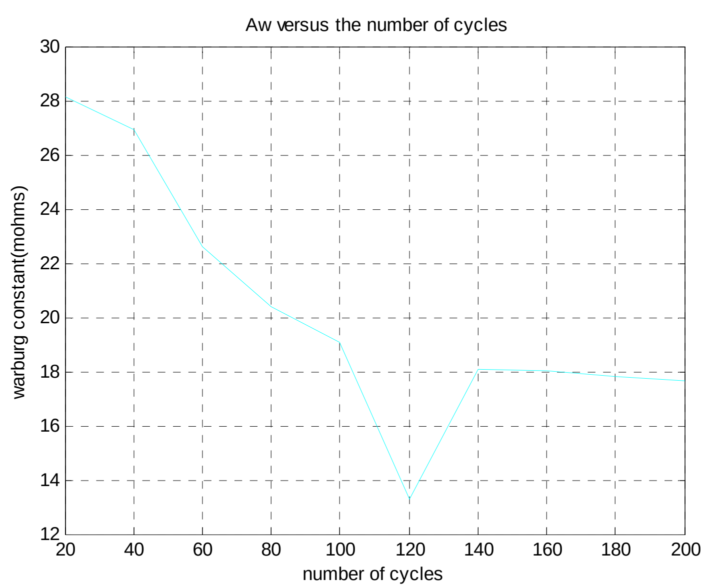


Figure 48. A_w Without excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

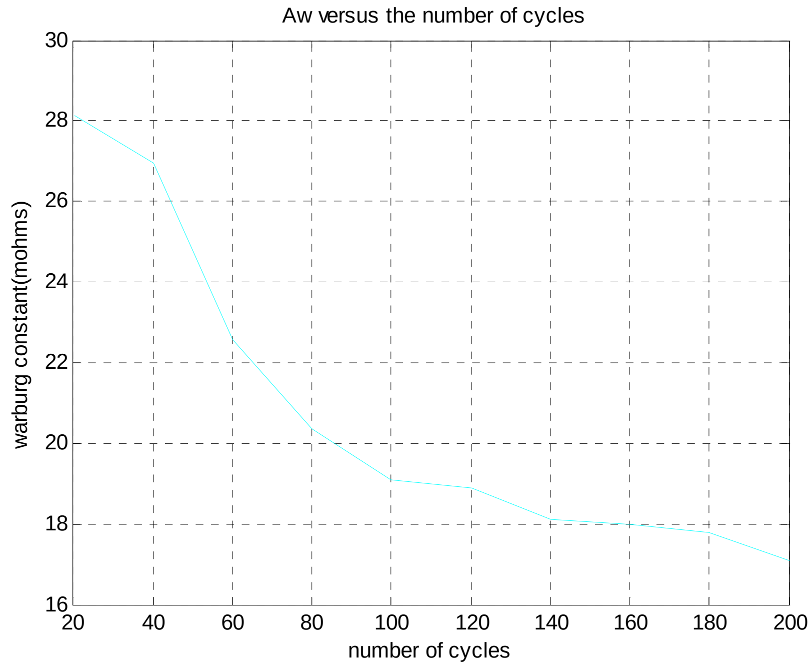


Figure 49. A_w With excluding the 3.3 Hz in loop (6) i.e. 120 cycles& 4.7Hz in loop (2) i.e. 40 cycles

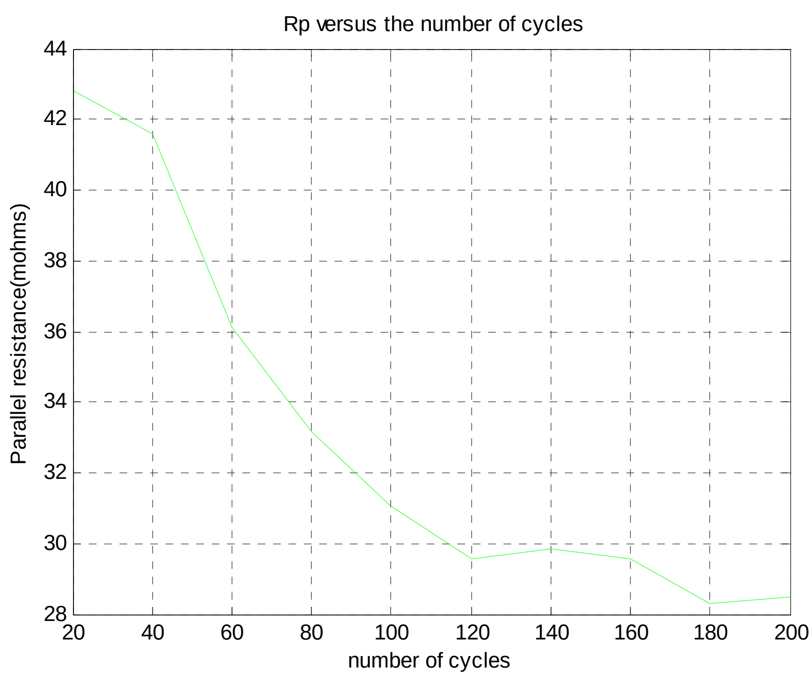


Figure 50. R_p Without excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

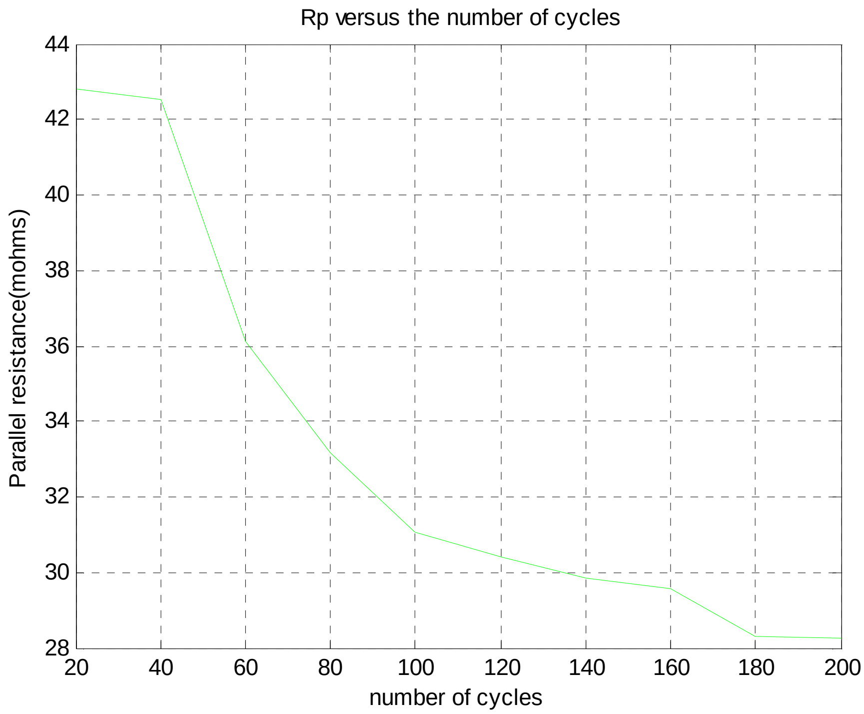


Figure 51. R_p With excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

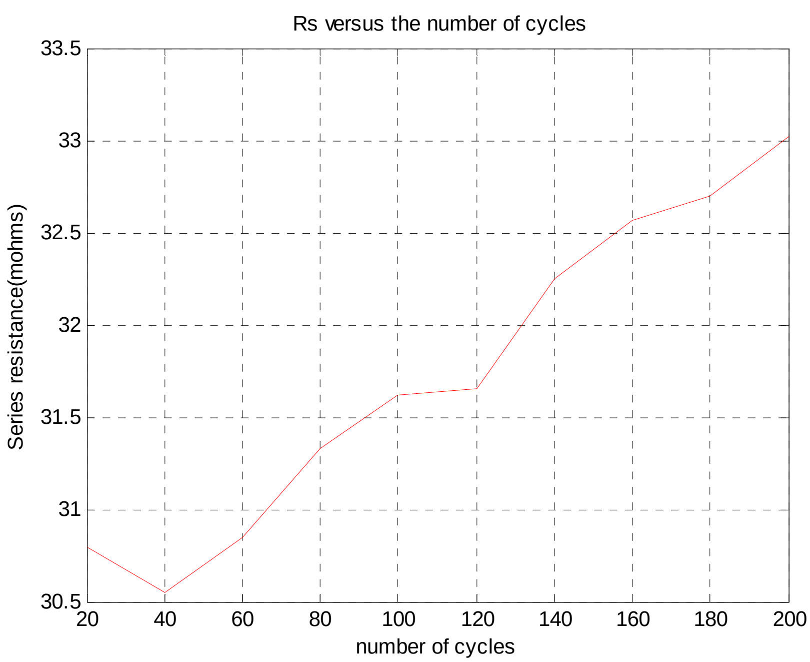


Figure 52. R_s Without excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

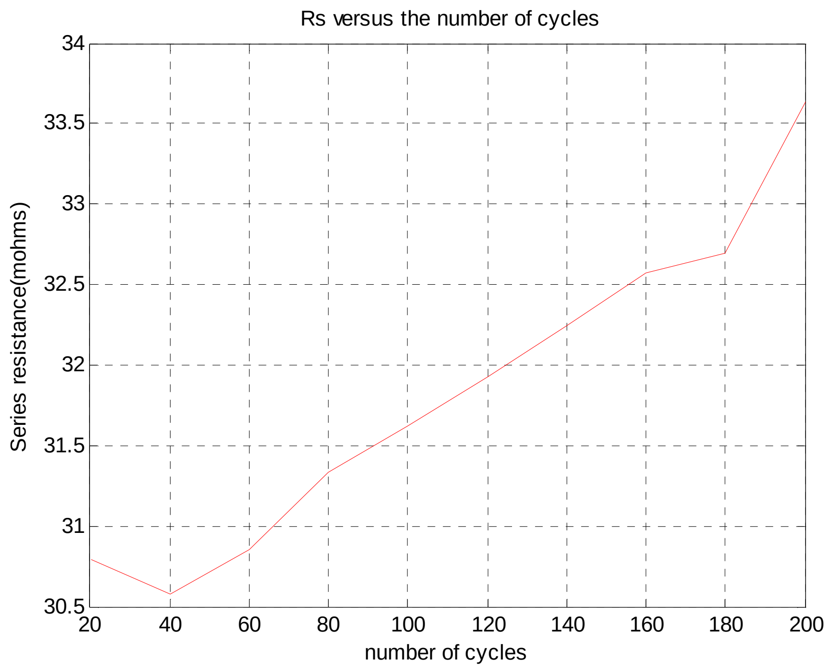


Figure 53. R_sWith excluding the 3.3 Hz in loop (6) i.e. 120 cycles & 4.7Hz in loop (2) i.e. 40 cycles

From above figures we can come up with the following observations, the Warburg constant A_w shows a decline manner, C_p (double layer capacitance) increase as the battery ages, R_p (the polarization resistance) has a dwindling trend and R_s (electrolyte resistance) rises with a slow rate although it still shows some irregular manner.

At a closer inspection of the data measured, we found out after investigating the real impedance and the magnitude of the impedance are compatible and have the same trend, this leads us to a conclusion that the real part of the impedance is the parameter which play the main role in the magnitude of the impedance of the battery as it ages. The real impedance of the battery and the magnitude of Z as the battery ages will be found in the appendix E, but it is noticeable that the variation of the values of real impedance as well as the magnitude with age is not showing a clear trend there is simply a very slight increase. It can be considered for further study and investigation since they have an increasing trend over a limited range. This is obvious in Figure 54and Figure55

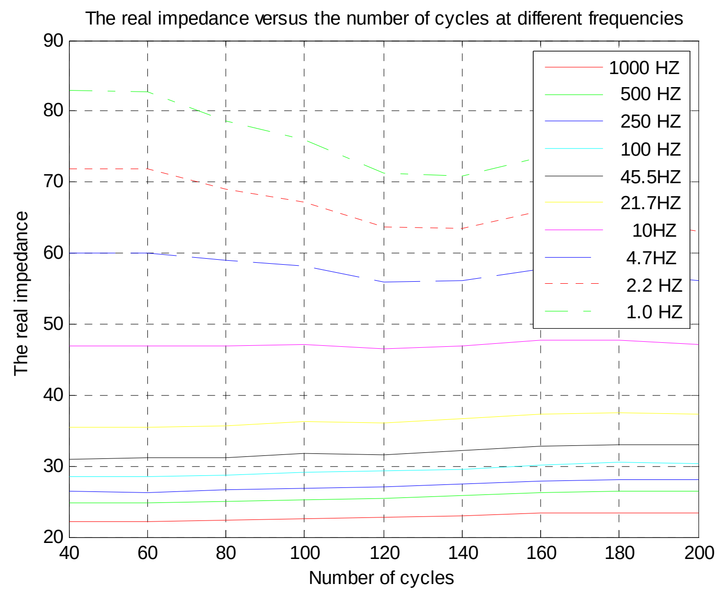


Figure 54. The real impedance versus the number of cycles for the first battery

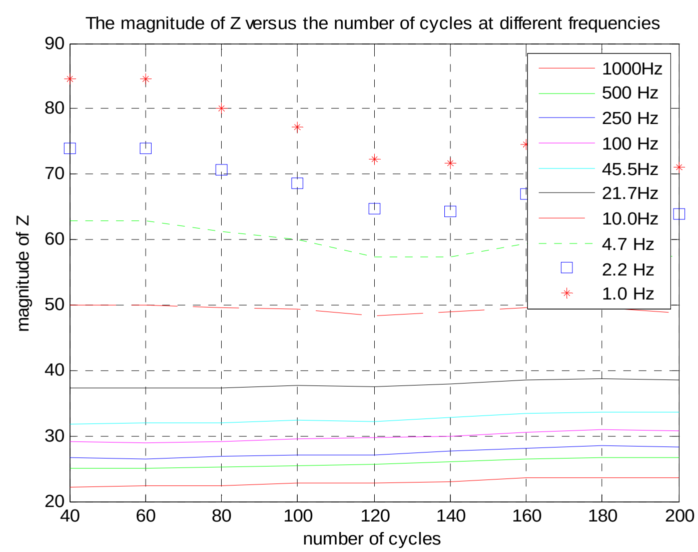
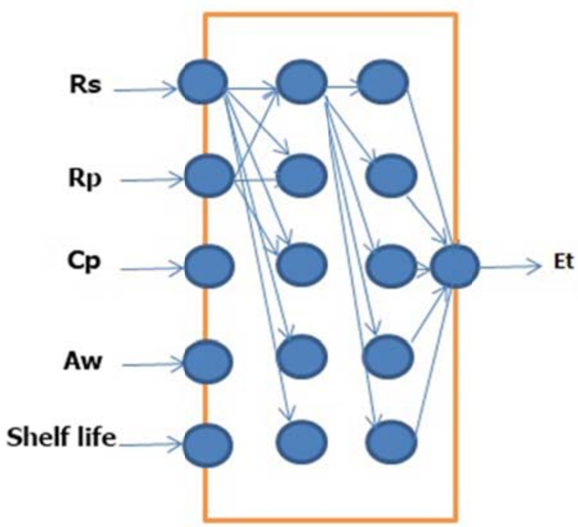


Figure 55. The magnitude of Z Versus the no. of cycles

D. Artificial Neural Network and Support Vector Regression

At the end we tried to promote the study to relate the Warburg infinite parameters extracted that best fit the battery model to the total energy delivered severally by using the Artificial Neural Network since we considered the energy delivered by each battery as an interpretation of its life time. We used two hidden layers with 5 neurons each. The reason for using only two hidden layers although a more precision is attained with an increasing number of hidden layers is that the size of the data set is relatively small that may cause an over-fitting. Figure 56 shows that the input data is the normalized values of R_s , R_p , C_p, A_w and the shelf life, recall equation (12), and the output is the normalized value of the total energy delivered each 20 cycles through the battery's life time to end up with the total energy delivered as the battery ages. This is carried out for the three samples.



ers of 5

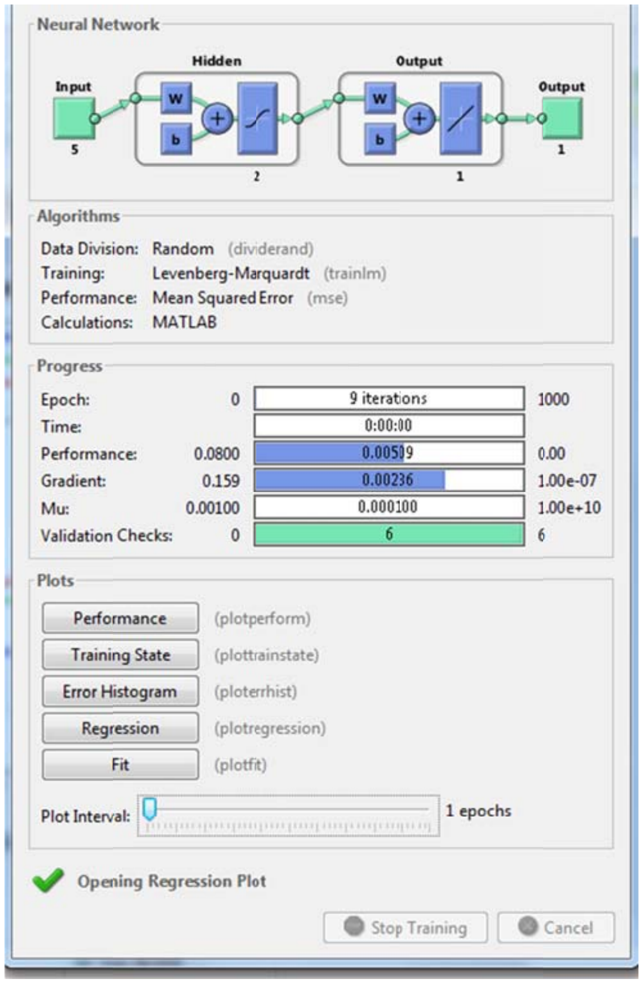


Figure 57. ANN used for training and validating the data

The performance plot in figure 58, shows that with 9 iterations only, using the gradient descent back propagation. The division for training, validating and testing the data was 70%, 15%, 15% successively randomly selected. The best validation performance is 0.0023 occurred at epoch 3 with mean square error equals 0.0067. The error histogram shown in figure 59, that gives uncertainties of the output data and the desired output for the training set and for the set for validation and testing. The regression plot in figure 60 shows that the regression for training is 92.4% while for validation is 99.9%, that for testing is 98.3% and the overall regression is 92.3%, it is known as the regression curves inclined by an angle 45° with the x-axis, the training is considered so far good and expressive. A table of the output and the desired output with the errors are included in appendix D.

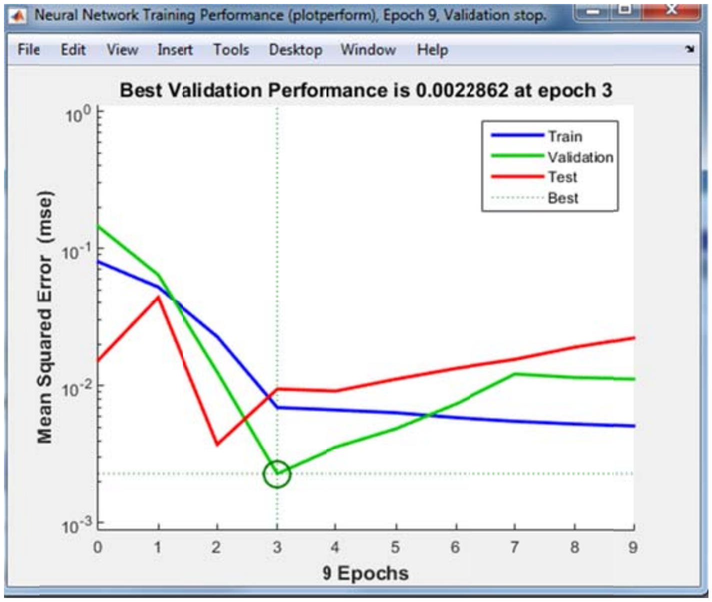


Figure 58. The performance plot

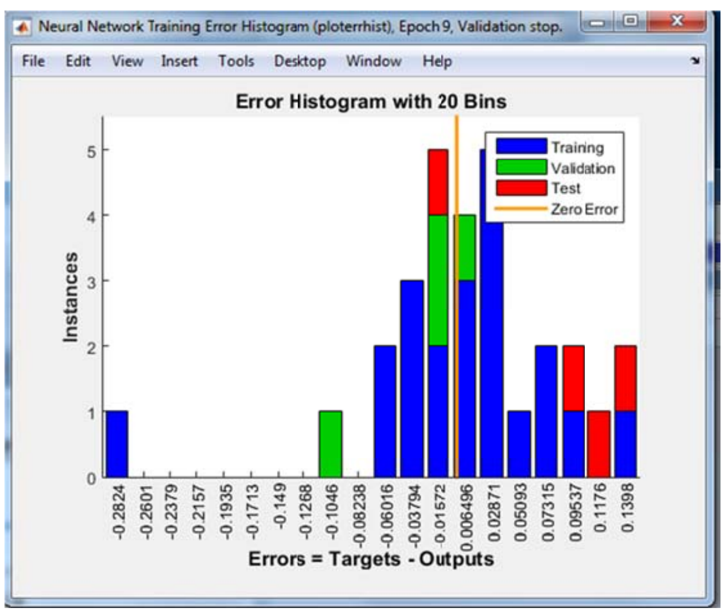


Figure 59. The error histogram

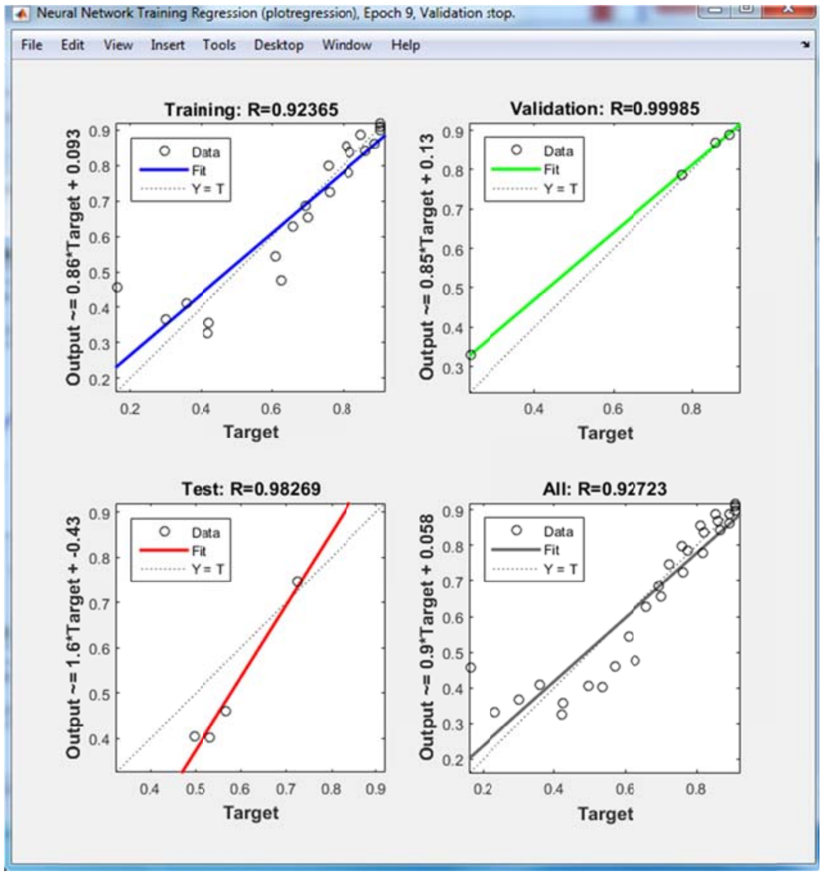


Figure 60. The regression plot

A closer look, it can be noticed that although the mean square error = 0.0067 but 6 validation failures had occurred, and the performance plot denotes that an over fitting happened, this is clear in figure 58. This happened because the data set size was relatively small. The validation is the measure of the performance of ANN.

Retrain the neural network again, we got a satisfactory results like

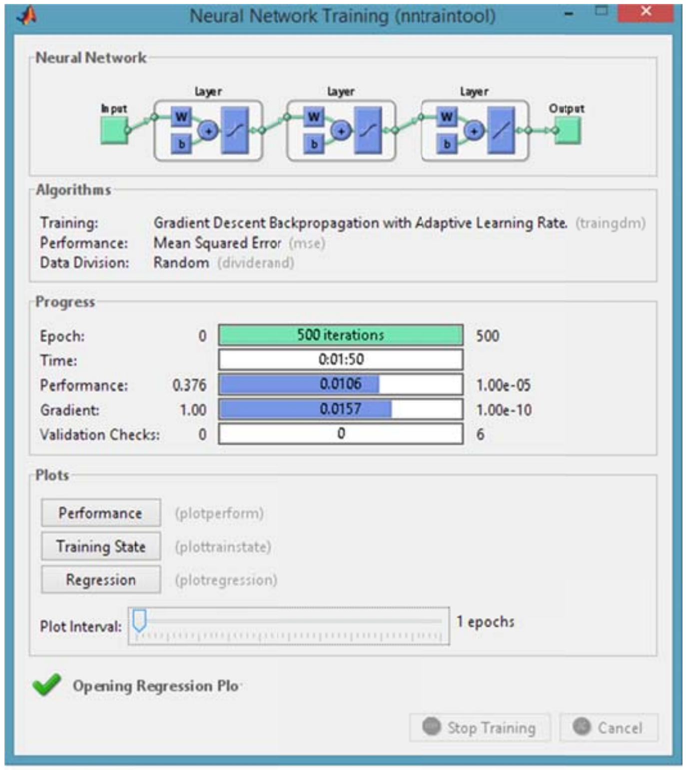


Figure 61. Retrain the ANN with no validation failures.

The performance plot in figure 62, shows that over fitting is minimized with no validation failures as seen in Figure 61, and the regression plot in figure 63 shows that the training was good and acceptable while the validation and testing regressions have deviations but the overall regression is good. The mean square error is 0.0106

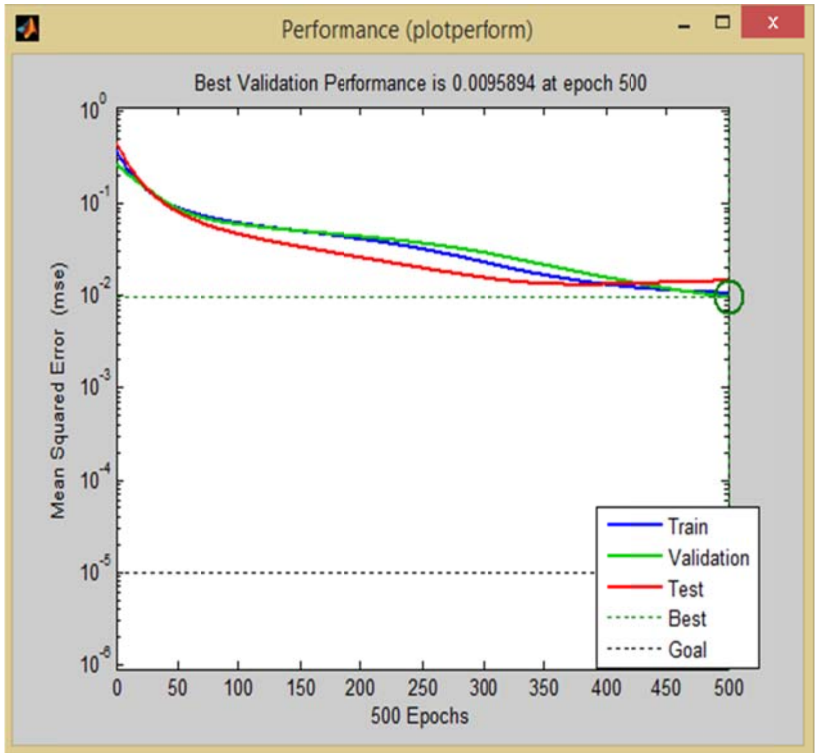


Figure 62. The performance plot for retraining the ANN

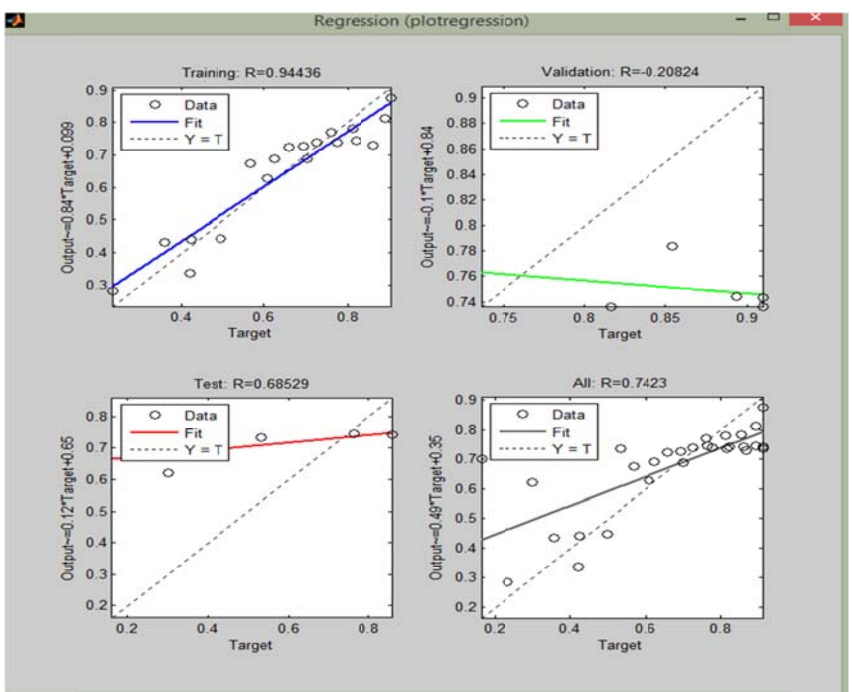


Figure 63. The regression plot for retraining

A more accurate training method is the support vector regression. A Gaussian SVR with 5 folds cross validation to train the given data using the MATLAB function **MdlGau=fitrsvm(data,y,'Standardize',true,'KFold',5,'KernelFunction','gaussian')** Is applied. This kind of regression is used to train a SVR model using Gaussian kernel with an auto scale. With 5 folds cross validation with the function **mseGau = kfoldLoss(MdlGau)**. The mean square error was equal to 0.0189.

Then a linear SVR is used to train the data using the above functions but replacing the Gau by Lin, in the cross validation function and Gaussian by linear in the "fitrsvm" function given above. The value of the mean is even better and equals 0.0098. We conclude from above that the Neural Network trained the data in a very good way but limited by the size of the data set but the support vector regression has the Excellency to train the data with the least mean square error by linear Support Vector Regression.

Tables below show the results of the ANN retraining process and the support vector regression, the results of the ANN training with over fitting is found in Appendix IV at the end of the thesis. We can see that the error between the desired E_{tot} as an interpretation of the battery life time and the output of training ANN and SVR is relatively small. But error has a greater value when the battery is new and at the beginning of the charging- discharging process as seen in the data with bolded red color.

The first red row is the first discharging process for the first battery, the second red row is the first discharging process for the second battery, and the third one is for the third battery. The SVR error for the first discharging process for the three samples is relatively smaller than that for the ANN.

ANN retraining			SVR training		
E_{tot}	output	Error	E_{tot}	Output	Error
0.359	**0.553**	**0.194**	**0.359**	**0.355**	**0.004**
0.496	0.562	0.066	0.496	0.515	0.019
0.609	0.5791	0.03	0.609	0.612	0.003
0.694	0.613	0.081	0.694	0.688	0.006
0.76	0.578	0.182	0.76	0.764	0.004
0.812	0.613	0.199	0.812	0.813	0.001
0.854	0.8863	0.032	0.854	0.871	0.017
0.892	0.872	0.02	0.892	0.877	0.015
0.909	0.973	0.064	0.909	0.926	0.017
0.235	**0.587**	**0.352**	**0.235**	**0.298**	**0.063**
0.42	0.577	0.157	0.42	0.430	0.010
0.568	0.61	0.042	0.568	0.568	0.000
0.66	0.632	0.028	0.66	0.663	0.003
0.725	0.655	0.07	0.725	0.728	0.003
0.776	0.698	0.078	0.776	0.772	0.004
0.82	0.78	0.04	0.82	0.835	0.015
0.859	0.864	0.005	0.859	0.871	0.012
0.893	0.86	0.033	0.893	0.877	0.016
0.909	0.975	0.066	0.909	0.928	0.019
0.162	**0.473**	**0.311**	**0.162**	**0.412**	**0.250**
0.301	0.388	0.087	0.301	0.348	0.047
0.424	0.42	0.004	0.424	0.501	0.077
0.533	0.367	0.166	0.533	0.530	0.003
0.626	0.456	0.17	0.626	0.746	0.0120
0.702	0.517	0.185	0.702	0.721	0.019
0.663	0.625	0.038	0.663	0.630	0.033
0.816	0.688	0.128	0.816	0.873	0.057
0.865	0.851	0.014	0.865	0.873	0.008
0.909	0.821	0.088	0.909	0.977	0.068

Table 1. Results of training ANN and SVR for the three lead-acid samples

In conclusion, three approaches were attempted for life time estimation of lead-acid battery of 7 Ah, 12 volts. First to determine if the battery is aged, the discharge time is used. If after 20 minutes we measure the voltage across the battery terminals and find out that the voltage drops to a value less than 1.75 V/ cell i.e. 10.5 V, then the battery should be replaced, otherwise it can still be used. This method helps us determine if the battery should be replaced or it is still useful, but it can't provide us with the life time of the battery.

The second approach relies on the energy integration process that can be evaluated using the V-t discharge curves for the battery all over its life time. The total energy delivered by the battery is evaluated by integrating the voltage discharge curve. It is dependent on the time that is spent over the shelf, which is hard to be found. So a combination of the two methods (energy integration and the time of discharge method) would tell the age of the battery based on the 5240 KWh ability of the new battery. It should be accompanied by a test to be run on the battery every once in a while from which one can tell that the battery has been on the shelf or has some manufacturing defect and if the battery is to be replaced or not.

The third approach is battery modeling. Many models were attempted like single Randle Cell, Double Randle Cell, Warburg impedance models, and Shepherd equation model. The condition to accept the model as best modeling the battery is fitting the measured real and imaginary impedances (and the terminal voltage for the Shepherd equation model), and the parameters show a clear pattern. Warburg infinite impedance shows an acceptable fit and for a range of frequency from 1 – 100 Hz and by excluding the frequencies that cause noises and the ill- mannered trends, it proved the best to model this type of batteries. After extracting the Warburg infinite parameters, we apply the

Artificial Neural Network for these parameters accompanied with the time spent on shelf to train the data and consider the energy delivered by the battery as an interpretation of the life time of the battery. A very good result obtained using the ANN training algorithm. The support vector linear regression gives even a better result with a mean square error = 0.0098 compared to the mean square error using Artificial Neural network that was = 0.0106.

There was a limitation when using the Artificial Neural Network, only a two hidden layers with 5 neurons each were used because the size of the data were small, to avoid over fitting. While the support vector regression is applied for all data sizes and it shows a fast and robust training process. Artificial Neural Network can hang up in a local minimum value that makes the training slow, however the Support Vector Regression has a unique solution.

E. Future Work

The study can be extended to include more battery samples to extract their parameters and study the variation of these parameters under different conditions like testing batteries at different higher temperatures using ovens, and different discharge currents. Also it is recommended to take more (EIS) readings for battery parameters, for example taking readings every 5 cycles rather than 20 cycles. Another thing is the ability to do (EIS) measurements at lower frequencies like 0.1 Hz.

REFERENCES

1) Oliver Tremblay, Louis-A. Dessaint, and Abdel-illah Dekkiche, “A Generic Battery Model for the Dynamic Simulation of Hybrid Electric Vehicles”, IEEE Vehicle Power and Propulsion Conference, 2007. Arlington, TX , 9-12 Sept. 2007, Page(s):284 - 289

2) Joya Zeitouny, Jihad Bou Merhi, Jana Chalak, Sami Karaki, “Estimation of Battery Internal Parameters”, IEEE International Conference on Industrial Technology (ICIT), 17-19 March 2015, Seville.

3) Mohamed Daowd, Noshin Omar, Bavo Verbrugge, Peter Van den Bossche, Joeri Van Mierlo, “Battery Models Parameter Estimation based on Matlab/ Simulink”, The 25th World Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exhibition, Nov. 5-9, 2010, EVS-25 Shenzhen, China,

4) Shuhui Li, Bao Ke, “Study of Battery Modeling using Mathematical and Circuit Oriented Approaches”, IEEE Power and Energy Society General Meeting, 24 - 29 July 2011, San Diego, CA, Pages 1-8.

5) Low Wen Yao, Aziz J.A, Pui Yee Kong, Idris N.R.N, “Modeling of Lithium – Ion Battery using Matlab / Simulink”, 39th Annual Conference of the IEEE Industrial Electronics Society (IECON) Nov. 10-14, 2013, Vienna, Austria,

6) Hanlei Zhang, Mo–Yuen Chow, “Comprehensive Dynamic Battery Modeling for PHEV Applications”, IEEE Power and Energy Society General Meeting, 25–29 July 2010, Minneapolis, MN, pages 1-6 .

7) M.R Jongerder and B.R Harverkot, “Battery Modeling”, Technical Report, TR-CTIT-08-01, Centre for Telematics and Information Technology, University of Twente, Enschede, 2008.

8) C. Unterrieder , C. Zhang , M. Lunglmayr , R. Priewasser , S. Marsili , M. Huemer “Battery state-of-charge estimation using approximate least squares”, *Journal of Power Sources*, Vol. 278, 15 March 2015, Pages 274–286

9) Toufik Madani Layadi, Gérard Champenois , Mohammed Mostefai, Dhaker Abbes, “Lifetime estimation tool of lead–acid batteries for hybrid power sources design” *Simulation Modelling Practice and Theory* ,Vol. 54, May 2015, Pages 36–48

10) David C. C. Freitas, Marcos B. Ketzer, Marcos R. A. Morais, Antonio M. N. Lima,“Life Estimation Techniques For Lead –Acid Batteries”, Industrial electronics society, IECON 2016-42nd Annual conference of IEEE

11) E. Jensi Miriam, S. Sekar, S. Ambalavanan, “Artificial Neural Network Technique for predicting the Lifetime and performance of Lead – Acid Battery”,

IRACST-An International Journal (ESTIJ), ISSN:2250-3498, Vol 3, No.2,April 2013.

12) Christopher Suozzo, “Lead-Acid Battery Aging And State Of Health Diagnosis”, A thesis for the Degree Master of Science in the graduate school of the Ohio State University, 2008.

13) http://www.mpoweruk.com/soh.htm, Electropedia “Battery and Energy Technologies”

14) B.S. Bhangu. P. Bentley, D.A. Stone and C.M. Bingham, “state-of-charge and state-of-health prediction of Lead-Acid batteries for Hybrid Electric Vehicles using non-linear observers”, IEEE, power electronics and applications, 2005 European conference.

15) V. Spath, A. Jossen, H. Doring, J. Garche, “ The detection of the state of health of Lead-Acid batteries”, IEEE, telecommunications Energy Conference,1997.

16) Kwang Kim, “ Electrochemical methods: Fundamentals and Applications” Lecture note 7, Younsi University, (2014).

17) Gamry Instruments. (2013). Application Note – Basics of Electrochemical Impedance Spectroscopy [Online]. Available https://www.gamry.com/application-notes/EIS/basics-of-electrochemical-impedance-spectroscopy/

18) Anton Anderson, “Battery Lifetime Modeling” Riso National Laboratory, Roskidle, Denmark, Rep, Riso – I – 2440, January 2006

19) S. Theodoridis and K. Koutroumbas, Pattern Recognition, Elsevier, USA, 2009 PP151-169 &PP 198 - 215

20) H. Bindner et al., “*Lifetime Modelling of Lead Acid Batteries*”, Riso National Laboratory,Roskilde, Denmark, Rep. Riso-R-1515(EN), 2005.

21) J.Leao,L. Hartmann, M.Correa, A.Lima, “ *Lead –acid battery modeling and state of charge monitoring*”, Applied Power Electronics conference and Exposition, APT), 25th Annual IEEE conference, pp 239-243, 2010

22) M.A.Casacca, Z.M.Salameh, “*Determination of lead –acid battery capacity via mathematical modeling techniques*”, IEEE transitions on energy coversion, Vol. 7, no.3, pp 442-446. 1992

23) Stephen Buller, “ *mpedance –Based Simulation Modelsof Supercapacitors and Li-Ion Batteries for power Electronic Applications”,* IEEE transactions on Industry Applications,Vol 41 NO.3,May/July 2005,pp742-746

APPENDICES

APPENDIX I

C Code

C code that drive the Programmable Power Supply, Electronic Load and BRS

```
#include "visa.h"
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include  <fcntl.h>
#include <string.h>
void discharge(float, float, FILE*);
void charge(float, float, float, double, FILE*);
void printvolt_currEL(ViSession, ...);
void printandsleep(double, FILE*); \
void printvolt_currPS(ViSession, ...);
void DisplayBRS(FILE*);
void delay(unsigned int);
float voltread = 0;
float currread = 0;
int main()
{
  time_t mytime;
        FILE *file;
        file = fopen("Data.txt", "w");
        if (file == NULL) {
                printf("Error opening the file \n");
                return(0);
        }
        //printf("Beginning of first time charge \n");
```

```
//fprintf(file, "Beginning of first time charge \n");

//charge(14.4, 1.75, 0.175, 0, file); //use the power supply to charge the battery arg1 is the voltage to be reached arg2 is the constant current at which charging occurs arg3 is the current decreasing threshold after the constant voltage mode is entered arg4 is the duration in hours that the supply must stay on after the current reaches the threshold

//printandsleep(0.25, file); //do nothing in 0.25 hourswhile printing the value on the screen

//DisplayBRS(file);

//printf("End of first time charge \n");

//fprintf(file, "End of first time charge \n");

for (int i = 0; i < 20; i++)

{

    printf("The loop is the number %d loop\n", i + 1);

    //printf("hola");

    fprintf(file, "loop is the %d \n", i + 1);

    for (int j = 0; j < 20; j++)

    {

    printf("The loop is the number %d loop  \n", j + 1);

        fprintf(file, "The loop is the number %d loop\n", j + 1);

charge(14.4, 1.75, 0.175, 0, file); //use the power supply to charge the battery arg1 is the voltage to be reached arg2 is the constant current at which charging occurs arg3 is the current decreasing threshold after the constant voltage mode is entered arg4 is the duration in hours that the supply must stay on after the current reaches the threshold= 0.175 A

        discharge(1.75, 10.5, file); //use the electronic load for discharging arg1 is the cosntant current at which the discharge happens arg2 is the threshold voltage=10.5 V                                        //printandsleep(1, file); //do nothing in hours while printing the value on the screen

                                        //charge(14.4, 1.75, 0.175, 0, file);//use the power supply to charge the battery arg1 is the voltage to be reached arg2 is the constant current at which charging occurs arg3 is the current decreasing threshold after the constant voltage mode is entered arg4 is the duration in hours that the supply must stay on after the current reaches the threshold

//discharge(1.75, 10.5, file);//use the electronic load for discharging arg1 is the cosntant current at which the discharge happens arg2 is the threshold voltage
```

```
	//printandsleep(1, file); //do nothing in hours while printing the value on the screen

										//charge(14.4,
1.75, 0.175, 0, file);//use the power supply to charge the battery arg1 is the voltage to be
reached arg2 is the constant current at which charging occurs arg3 is the current decreasing
threshold after the constant voltage mode is entered arg4 is the duration in hours that the
supply must stay on after the current reaches the threshold

	//discharge(1.75, 10.5, file);//use the electronic load for discharging arg1 is the
cosntant current at which the discharge happens arg2 is the threshold voltage

	//printandsleep(1, file); //do nothing in hours while printing the value on the screen

		}
		charge(14.4, 1.75, 0.175, 0, file);//use the power supply to charge the battery
arg1 is the voltage to be reached arg2 is the constant current at which charging occurs arg3 is
the current decreasing threshold after the constant voltage mode is entered arg4 is the
duration in hours that the supply must stay on after the current reaches the threshold
		printandsleep(0.25, file); //do nothing in hours while printing the value on the
screen
		DisplayBRS(file);
	}
	printf("Finishing up the program. Closing the file \n");
	fprintf(file, "Finishing up the program. Closing the file \n");
	fclose(file);
	return 0;
}

void charge(float voltsetting, float currsetting, float currthresh, double duration, FILE *file) {
	time_t mytime;
	printf("Start Charging process at the following time: \n");
	fprintf(file, "Start Charging process at the following time: \n");
	mytime = time(NULL);
```

```
        printf(ctime(&mytime));
        fprintf(file, ctime(&mytime));
    ViSession session1, defrm1, session2, defrm2;
        ViStatus VISAstatus1, VISAstatus2;
        char statdsc[100];
        char err[100];
        char voltmeasurement[32] = "0";
        int overcurron;
        float  overvoltsetting;
        char add1[] = "GPIB0::4::INSTR";
        char add2[] = "GPIB0::5::INSTR";

        //these vaiables control the over voltage protection setting
        overvoltsetting = 20;  //in volts

                                    //this variable controls the over current
protection (1 for on, 0 for off);
        overcurron = 0;
        // connection to DC POWER Supply

        //The default resource manager manages initializes the VISA system
        VISAstatus1 = viOpenDefaultRM(&defrm1);
        if (VISAstatus1 != VI_SUCCESS)
        {
                viStatusDesc(defrm1, VISAstatus1, statdsc);
                printf("Error on viOpen: %s \n", statdsc);
        exit(EXIT_FAILURE);
        }
        //opens a communication session with the instrument at address "add1"
        VISAstatus1 = viOpen(defrm1, add1, VI_NULL, VI_NULL, &session1);
```

```
if (VISAstatus1 != VI_SUCCESS)
{
        viStatusDesc(session1, VISAstatus1, statdsc);
        printf("Error on viOpen: %s \n", statdsc);
        printf("Error communicating with power supply");
        exit(EXIT_FAILURE);
}

//connection to the electronic load

VISAstatus2 = viOpenDefaultRM(&defrm2);
if (VISAstatus2 != VI_SUCCESS)
{
        viStatusDesc(defrm2, VISAstatus2, statdsc);
        printf("Error on viOpen: %s \n", statdsc);
        exit(EXIT_FAILURE);
}

//opens a communication session with the instrument at address "add2"
VISAstatus2 = viOpen(defrm2, add2, VI_NULL, VI_NULL, &session2);
if (VISAstatus2 != VI_SUCCESS)
{
        viStatusDesc(session2, VISAstatus2, statdsc);
        printf("Error on viOpen: %s \n", statdsc);
        printf("Error communicating with Electronic Load");
        exit(EXIT_FAILURE);
}
viPrintf(session2, "CHAN 1;:INPUT OFF \n");

printf("Voltage and current initially are: \n");
```

```
fprintf(file, "Voltage and current initially are: \n");
printvolt_currPS(session1, voltread, currread);
fprintf(file, "Voltage: %f     Current: %f \n", voltread, currread);

//Set voltage
viPrintf(session1, "VOLT %f \n", voltsetting);
//Set the over voltage level
viPrintf(session1, "VOLT:PROT:LEV %f \n", overvoltsetting);

//Turn OFF over current protection
viPrintf(session1, "CURR:PROT:STAT %d \n", overcurron);

//Set current level
viPrintf(session1, "CURR %f \n", currsetting);
Sleep(500);
//Enable the output
viPrintf(session1, "OUTP ON \n");
Sleep(500);
printvolt_currPS(session1, voltread, currread);
fprintf(file, "Voltage: %f     Current: %f \n", voltread, currread);
while (voltread < voltsetting) {
        printvolt_currPS(session1, voltread, currread);
        fprintf(file, "Voltage: %f     Current: %f \n", voltread, currread);
        Sleep(60000);
}

printf("Voltage Reached. Done Charging. Monitoring current until threshold is reached
\n");
fprintf(file, "Voltage Reached. Done Charging. Monitoring current until threshold is
reached \n");
```

```
while (currread > currthresh) {
        printvolt_currPS(session1, voltread, currread);
        fprintf(file, "Voltage: %f      Current: %f \n", voltread, currread);
        Sleep(60000);
}
//printandsleep(duration, file);
printf("Done charging and Sleep to specified voltage \n");
fprintf(file, "Done charging and Sleep to specified voltage \n");
printf("turning OFF power Supply \n");
fprintf(file, "turning OFF power Supply \n");
viPrintf(session1, "OUTP OFF \n");
Sleep(1000);
//check for errors
viPrintf(session1, "SYST:ERR? \n");
Sleep(1000);
viScanf(session1, "%t", &err);
printf("Error: %s \n", err);
//This frees up all of the resources
viClose(session1);
viClose(defrm1);
//check for errors
viPrintf(session2, "SYST:ERR? \n");
Sleep(1000);
viScanf(session2, "%t", &err);
printf("Error: %s \n", err);
//This frees up all of the resources
viClose(session2);
viClose(defrm2);
printf("Done Charging process and topping. Starting the discharge process at this time:
\n");
```

```
	fprintf(file, "Done Charging process and topping. Starting the discharge process at this time: \n");
	mytime = time(NULL);
	printf(ctime(&mytime));
	fprintf(file, ctime(&mytime));

	return;
}

void discharge(float curr, float voltthreshhold, FILE *file)
{

	printf("Starting the discharge process with constant current of %f and voltage threshold of %f \n", curr, voltthreshhold);
	fprintf(file, "Starting the discharge process with constant current of %f and voltage threshold of %f \n", curr, voltthreshhold);
	ViSession session, defrm;
	ViStatus VISAstatus;
	char statdsc[100];
	char err[100];
	char voltmeasurement[32] = "0";
	char add[] = "GPIB0::5::INSTR";

	VISAstatus = viOpenDefaultRM(&defrm);
	if (VISAstatus != VI_SUCCESS)
	{
		viStatusDesc(defrm, VISAstatus, statdsc);
		printf("Error on viOpen: %s \n", statdsc);
		exit(EXIT_FAILURE);
	}
```

```
	//opens a communication session with the instrument at address "add"
	VISAstatus = viOpen(defrm, add, VI_NULL, VI_NULL, &session);
	if (VISAstatus != VI_SUCCESS)
	{
		viStatusDesc(session, VISAstatus, statdsc);
		printf("Error on viOpen: %s \n", statdsc);
		exit(EXIT_FAILURE);
	}

	viPrintf(session, "CHAN 1;:INPUT OFF \n");
	viPrintf(session, "FUNCTION CURRENT \n");
	viPrintf(session, "CURRENT:LEVEL %f \n", curr - 0.01); // 0.01 calibration
	viPrintf(session, "CHAN 1;:INPUT ON \n");

	printvolt_currEL(session, &voltread, &currread);
	fprintf(file, "Voltage: %f	Current: %f \n", voltread, currread);

	while (voltread > voltthreshhold) {
		printvolt_currEL(session, &voltread, &currread);
		fprintf(file, "Voltage: %f	Current: %f \n", voltread, currread);
		Sleep(60000);
	}



	viPrintf(session, "CHAN 1;:INPUT OFF \n");
	printf("Stopping the discharge process by eliminating the load of the Electronic Load
Machine \n");
	fprintf(file, "Stopping the discharge process by eliminating the load of the Electronic
Load Machine \n");
	viPrintf(session, "SYST:ERR? \n");
```

```
        Sleep(1000);
        viScanf(session, "%t", &err);
        printf("Error: %s \n", err);
        //This frees up all of the resources
        viClose(session);
        viClose(defrm);
        time_t mytime;
        printf("Done disCharging process . Time is: \n");
        fprintf(file, "Done disCharging process .  Time is: \n");
        mytime = time(NULL);
        printf(ctime(&mytime));
        fprintf(file, ctime(&mytime));
        return;
}
void printvolt_currEL(ViSession session, ...) {


        viPrintf(session, "MEASURE:VOLTAGE? \n");
        Sleep(1000);
        viScanf(session, "%f", &voltread);
        viPrintf(session, "MEASURE:CURRENT? \n");
        Sleep(1000);
        viScanf(session, "%f", &currread);
        printf("Voltage: %f      ", voltread);
        printf("Current: %f \n", currread);
}
void printvolt_currPS(ViSession session, ...) {
        viPrintf(session, "MEAS:VOLT? \n");
        Sleep(1000);
        viScanf(session, "%f", &voltread);
        viPrintf(session, "MEAS:CURR? \n");
```

```
        Sleep(1000);
        viScanf(session, "%f", &curread);
        printf("Voltage: %f      ", voltread);
        printf("Current: %f \n", currread);
}
void printandsleep(double duration, FILE *file) {
        \

                ViSession session, defrm;
        ViStatus VISAstatus;
        char statdsc[100];
        char err[100];
        char voltmeasurement[32] = "0";
        char add[] = "GPIB0::5::INSTR";
        VISAstatus = viOpenDefaultRM(&defrm);
        if (VISAstatus != VI_SUCCESS)
        {
                viStatusDesc(defrm, VISAstatus, statdsc);
                printf("Error on viOpen: %s \n", statdsc);
                exit(EXIT_FAILURE);
        }

        //opens a communication session with the instrument at address "add"
        VISAstatus = viOpen(defrm, add, VI_NULL, VI_NULL, &session);
        if (VISAstatus != VI_SUCCESS)
        {
                viStatusDesc(session, VISAstatus, statdsc);
                printf("Error on viOpen: %s \n", statdsc);
                exit(EXIT_FAILURE);
        }
```

```
	double d = duration * 3600;
	time_t start, end;
	printf("Sleeping for 0.25 hour. The voltage and current will be displayed for analytical purposes \n");
	fprintf(file, "Sleeping for 0.25 hour. The voltage and current will be displayed for analytical purposes \n");
	start = time(NULL);
	Sleep(1500);
	end = time(NULL);
	while (difftime(end, start) <= d) {
		printvolt_currEL(session);
		fprintf(file, "Voltage: %f	Current: %f \n", voltread, currread);
		Sleep(60000);
		end = time(NULL);
	}
	printf("Waking UP\n");
	fprintf(file, "Waking UP\n");
	viPrintf(session, "SYST:ERR? \n");
	Sleep(1000);
	viScanf(session, "%t", &err);
	printf("Error: %s \n", err);
	//This frees up all of the resources
	viClose(session);
	viClose(defrm);
```

```
	time_t mytime;
	printf("Done Sleeping for 0.25 of an hour at time: \n");
	fprintf(file, "Done Sleeping for 0.25 of an hour at time: \n");
	mytime = time(NULL);
	printf(ctime(&mytime));
	fprintf(file, ctime(&mytime));
	return;}
void DisplayBRS(FILE *f) {
	FILE *file;
	file = fopen("BIMresults.log", "r");
	char line[1024];
	if (file == NULL) {
		printf("error \n");
	}

	else
	{
		while (fgets(line, 1024, file))
		{
			if (line[91] == '<')
			{
				printf("BRS VALUES ARE--- \n");
				fprintf(f, "BRS VALUES ARE- \n");
				printf("Line: %s\n", line);
				fprintf(f, "Line: %s\n", line);
			}
		}
	}
	fclose(file);
```

```
}
void delay(unsigned int mseconds)
{
	clock_t goal = mseconds + clock();
	while (goal > clock());
}
```

APPENDIX II

Parameters extraction

1) The impedance calculation MATLAB code

```
function [Zr, Zm] = ImpedanceCalc(P, w, nexp, model)

% This function calculates the equivalent impedance of our
% battery circuit model (including the Warburg impedance).

if (model == 1)
    % Model Impedance by Randles Cell
    Rs = P(1);  % Rs (series resistance)
    Rp = P(2);  % Rp (transient resistance)
    Tp = P(3);  % Tp (transient time constant)
    Z= Rs + Rp./(1 + (1i* w).^nexp * Tp);

elseif (model == 2)

    % % Use a double Randles cell
    Rs = P(1);   % Rs (series resistance)
    Rp1 = P(2);  % Rp (transient resistance)
    Tp1 = P(3);  % Tp (transient time constant)
    Rp2 = P(4);  % Rp (transient resistance)
    Tp2 = P(5);  % Tp (transient time constant)
    Z= Rs + Rp1./(1 + (1i* w).^nexp(1) * Tp1) + Rp2./(1 + (1i*
w).^nexp(2) * Tp2);

elseif (model == 3)

    % % Model Impedance using Warburg
    Rs = P(1);  % Rs (series resistance)
    Rp = P(2);  % Rp (transient resistance)
    Cp = P(3);  % Cp (transient capacitance)
    Aw = P(4);  % Aw (Warburg impedance constant)
    Zw = (Aw./sqrt(w)).*(1-1i);              % Zw is the Warburg
impedance.
    Zp = ((1i.*w.*Cp)+1./(Rp + Zw)).^(-1);   % Zp the equivalent
impedance of the parallel branch.
    Z = Rs + Zp;

elseif (model == 4)
    % % Model Impedance using finite Warburg impedance in parralel
with Rp
    Rs = P(1);  % Rs (series resistance)
    Rp = P(2);  % Rp (transient resistance)
    Cp = P(3);  % Cp (transient capacitance)
    Aw = P(4);  % Aw (1st Warburg impedance parameter)
    Bw = P(5);  % Bw (2nd Warburg impedance parameter)

    % Zw is the finite Warburg impedance.
```

```
    Zw = (Aw./sqrt(w)).*(1-1i).* tanh(Bw* sqrt(1i*w));
    Zp = ((1i.*w.*Cp)+1./(Rp + Zw)).^(-1);   % Zp the equivalent
impedance of the parallel branch.
    Z= Rs + Zp;

end

Zr = real(Z);   % real part of equivalent impedance.
Zm = imag(Z);   % imaginary part of equivalent impedance.

end
```

2) **The sum of square error function MATLAB code**

```
function [J, GJ] = sumSquareErrors(P, Zr, Zm, w, nexp, model)

% This function calculates the sum of squares of errors between the
% experimental impedance value and that of the model. I t also
% calculates its gradient needed for optimization.

% Zrm and Zmm are the real and imaginary parts of the impedance
% as calculated by the model .
[Zrm, Zmm] = ImpedanceCalc(P, w, nexp, model);

% Performance index
J= (Zr- Zrm)'* (Zr- Zrm) + (Zm - Zmm)'* (Zm - Zmm);

nP= size(P,1);
GJ= zeros(nP,1);

% Assign disturbance values of 1 to 5%
DPO = P*0.05;

% Calculate gradient of J using a finite difference method
for k = 1 : nP

    % Initialize the disturbance vector.
    dP = zeros(nP, 1);

    % Assign a perturbation for the selected parameter (k).
    dP(k) = DPO(k)/2;

    [Zr1, Zm1] = ImpedanceCalc(P-dP, w, nexp, model);
    [Zr2, Zm2] = ImpedanceCalc(P+dP, w, nexp, model);

    % Compute element (k) of the gradient of J
    J1= (Zr- Zr1)'* (Zr- Zr1) + (Zm - Zm1)'* (Zm - Zm1);
    J2= (Zr- Zr2)'* (Zr- Zr2) + (Zm - Zm2)'* (Zm - Zm2);
    GJ(k)= (J2-J1)/ DPO(k);

end

end
```

3) The MATLAB code for the function "FminUnc"

This function minimizes the sum of square errors of the measured values to the model

```
% The following program is meant to calculate a minimal-error
approximation
% of a vector P which includes Rs, Rp, tp (= Rp x Cp), and Aw which
are the
% characteristic components of our battery circuit model.

clc
close all
format long
iterMax = 1;       % Maximum number of iterations
tol = 1e-6;         % The tolerance for error.

fact= 0.001; % Convert impedances to ohms

% Identify Models. Set model in data file.
modelName= [...
        'One Randles Cell with exponent                  ';
        'Two Randles Cell with exponent                  ';
        'A Randles Cell with infinite Warburg impedance ';
        'A Randles Cell with finite Warburg impedance   ' ];

% Small set of data to test
% Zr=  fact* [ 1.618, 1.638, 1.682, 1.725, 1.767, 1.827, 1.871 ]'
% Zm= -fact* [ 0.031, 0.066, 0.116, 0.151, 0.170, 0.182, 0.187 ]'
% w=   2*pi* [ 333,   250,   143,   100,   71.4,  45.5,  33.3  ]'

% Get Data

% Lithium_001_40soc_23p5d         % lithium ion
% Lithium_001_40soc_23p7d         % lithium ion
% Lithium_001_40soc_24p1d         % lithium ion
% Lithium_001_40soc_24p2d         % lithium ion
% Lithium_001_40soc_24p3d         % lithium ion
% Lithium_001_40soc_24p5d         % lithium ion
% Lithium_001_40soc_24p6d         % lithium ion
% Lithium_001_40soc_24p7d         % lithium ion

% Lithium_001_40soc_10d           % lithium ion

% Lead_Acid_007_100soc_34d            % lead acid
% Lead_Acid_007_3_100soc_7p5d           % lead acid
% Lead_Acid_007_3_2_100soc_7p5d           % lead acid
% Lead_Acid_007_3_0_100soc_7p5d
% Lead_Acid_007_3_1_100soc_7p5d
Lead_Acid_007_3_1_100soc_7p5d

Zr= fact* Data(:,2);
Zm= fact* Data(:,3);
w= 2*pi* Data(:,1);

% Minimize using WLS pseudo-inverse
```

```
% [P, flag]= minSumSquares(P0, Zr, Zm, w, nexp, model, 20, 1e-12);

% Unconstrained minimization using MATLAB function
% options = optimset('GradObj','on','TolX', 1e-12,'TolFun',1e-12,
'MaxIter',100 ); % gradient provided
options = optimset('GradObj','on','TolX', 1e-12,'TolFun',1e-
12,'MaxIter',800,'MaxFunEvals', 800);  % gradient provided
% [P,fval,exitFlag] = fminunc(@(P) sumSquareErrors(P, Zr, Zm, w, nexp,
model), P0, options);
[P,fval,exitFlag] = fminunc(@(P) sumSquareErrors2(P, Zr, Zm, w, nexp,
model), P0, options);
% sumSquares= fval;

% Evaluate
[Zrm, Zmm] = ImpedanceCalc(P, w, nexp, model);
sumSquares= sum( ((Zr-Zrm)).^2) + sum( ((Zm-Zmm)).^2);

fprintf('\n')
fprintf(['---------- Summary of Results: ',bat.Type,' ----------
\n\n'])
fprintf('Battery State of Charge (percent)=   %9d\n', bat.SOC* 100);
fprintf('Battery Temperature (C)=             %9.2f\n', bat.Temp );
fprintf(['Model Name:  ',modelName(model,:),'\n'])
fprintf('Model exponent (n)=                    %9.3f\n', nexp);
fprintf('Sum of Squared Errors=                 %0.6g\n', sumSquares);
fprintf('\n')

writeResults

plotResults
```

APPENDIX III

ANN Matlab code

The MATLAB code for the ANN training algorithm

```
% Battery parameters

data= [ ...
%  Rsn   Rpn     Cpn     Awn     SL
0.1371  0.4508  0.3750  0.3666  0.1930
0.1211  0.4612  0.3684  0.3367  0.1930
0.1139  0.3231  0.3684  0.2588  0.1930
0.1898  0.2394  0.4145  0.1808  0.1930
0.1298  0.0943  0.4671  0.1335  0.1930
0.2521  0.0766  0.4868  0.1064  0.1930
0.5066  0.1269  0.5066  0.1135  0.1930
0.4776  0.1332  0.6184  0.0501  0.1930
0.8054  0.0000  1.0000  0.0463  0.1930
0.0541  0.5355  0.1447  0.3901  0.2900
0.0000  0.5259  0.1645  0.3487  0.2900
0.0670  0.2954  0.1908  0.1939  0.2900
0.1835  0.1906  0.2500  0.1159  0.2900
0.2536  0.1154  0.2895  0.0705  0.2900
0.3285  0.0922  0.3355  0.0630  0.2900
0.4049  0.0719  0.3421  0.0358  0.2900
0.4830  0.0606  0.3421  0.0324  0.2900
0.5144  0.0156  0.3947  0.0252  0.2900
0.7409  0.0147  0.5461  0.0000  0.2900
0.6000  0.8192  0.2171  0.8392  0.0473
0.4931  1.0000  0.0395  1.0000  0.0473
0.3244  0.9819  0.0000  0.9555  0.0473
0.6749  0.9744  0.0789  0.9790  0.0473
0.6193  0.8087  0.1908  0.8717  0.0473
0.6502  0.6386  0.3092  0.7518  0.0473
0.8313  0.6417  0.4342  0.6419  0.0473
0.9045  0.5786  0.5658  0.5993  0.0473
1.0000  0.5661  0.6118  0.4668  0.0473
0.9265  0.4344  0.6184  0.4324  0.0473

];

target= [...

0.359
0.496
0.609
0.694
0.760
0.812
0.854
0.892
```

```
0.909
0.235
0.420
0.568
0.660
0.725
0.776
0.820
0.859
0.893
0.909
0.162
0.301
0.424
0.533
0.626
0.702
0.763
0.816
0.865
0.909


];


x = data';
t = target';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 2;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean Squared Error
```

```
% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x)
e = gsubtract(t,y)
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code
blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

APPENDIX IV

ANN and SVR input data and results

1) The raw data used for ANN and SVR training process

R_s	R_p	C_p	A_w	SL	E_t
31.139	40.428	0.466	27.475	0.153	1752.4
31.073	40.718	0.465	26.628	0.153	2421.2
31.043	36.868	0.465	24.42	0.153	2973.6
31.357	34.534	0.472	22.21	0.153	3390.4
31.109	30.488	0.48	20.869	0.153	3711.5
31.615	29.994	0.483	20.101	0.153	3965.3
32.668	31.396	0.486	20.301	0.153	4172.1
32.548	31.572	0.503	18.505	0.153	4355.4
33.904	27.858	0.561	18.397	0.153	4440.5
30.796	42.79	0.431	28.139	0.256	1010.0
30.572	42.522	0.434	26.968	0.256	1803.2
30.849	36.094	0.438	22.58	0.256	2437.9
31.331	33.173	0.447	20.371	0.256	2832.7
31.621	31.077	0.453	19.084	0.256	3111.1
31.931	30.429	0.46	18.872	0.256	3330.3
32.247	29.863	0.461	18.1	0.256	3518.4
32.57	29.549	0.461	18.004	0.256	3686.4
32.7	28.292	0.469	17.801	0.256	3833.5
33.637	28.268	0.492	17.086	0.256	3901.8
33.054	50.7	0.442	40.865	0	937.1
32.612	55.742	0.415	45.423	0	1733.3
31.914	55.237	0.409	44.162	0	2443.9
33.364	55.029	0.421	44.829	0	3073.8
33.134	50.409	0.438	41.788	0	3611.1
33.262	45.665	0.456	38.389	0	4050.0
34.011	45.752	0.475	35.276	0	4401.6
34.314	43.993	0.495	34.069	0	4706.4
34.709	43.644	0.502	30.315	0	4988.8
34.405	39.971	0.503	29.338	0	5243.1

2) The normalized data set

R_{sn}	R_{pn}	C_{pn}	A_{wn}	SL	E_{tn}
0.137	0.451	0.375	0.367	0.193	0.359
0.121	0.461	0.368	0.337	0.193	0.496
0.114	0.323	0.368	0.259	0.193	0.609
0.190	0.239	0.414	0.181	0.193	0.694
0.130	0.094	0.467	0.134	0.193	0.760
0.252	0.077	0.487	0.106	0.193	0.812
0.507	0.127	0.507	0.113	0.193	0.854
0.478	0.133	0.618	0.050	0.193	0.892
0.805	0.000	1.000	0.046	0.193	0.909
0.054	0.536	0.145	0.390	0.29	0.235
0.000	0.526	0.164	0.349	0.29	0.420
0.067	0.295	0.191	0.194	0.29	0.568
0.183	0.191	0.250	0.116	0.29	0.660
0.254	0.115	0.289	0.071	0.29	0.725
0.328	0.092	0.336	0.063	0.29	0.776
0.405	0.072	0.342	0.036	0.29	0.820
0.483	0.061	0.342	0.032	0.29	0.859
0.514	0.016	0.395	0.025	0.29	0.893
0.741	0.015	0.546	0.000	0.29	0.909
0.600	0.819	0.217	0.839	0.0473	0.162
0.493	1.000	0.039	1.000	0.0473	0.301
0.324	0.982	0.000	0.955	0.0473	0.424
0.675	0.974	0.079	0.979	0.0473	0.533
0.619	0.809	0.191	0.872	0.0473	0.626
0.650	0.639	0.309	0.752	0.0473	0.702
0.831	0.642	0.434	0.642	0.0473	0.763
0.905	0.579	0.566	0.599	0.0473	0.816
1.000	0.566	0.612	0.467	0.0473	0.865
0.927	0.434	0.618	0.432	0.0473	0.909

3) Results from training the input data set

Etn	desired o/p	error	train perf.	val perf	test perf
0.359	0.4097	0.0507	0.007	0.0023	0.0095
0.496	0.4052	0.0908			
0.609	0.5440	0.0646			
0.694	0.6869	0.0071			
0.760	0.7994	0.0394			
0.812	0.8556	0.0436			
0.854	0.8876	0.0336			
0.892	0.8621	0.0299			
0.909	0.9190	0.01			
0.235	0.3295	0.0945			
0.420	0.3261	0.0939			
0.568	0.4592	0.1088			
0.660	0.6269	0.0331			
0.725	0.7458	0.0208			
0.776	0.7865	0.0105			
0.820	0.8368	0.0167			
0.859	0.8680	0.0092			
0.893	0.8891	0.0039			
0.909	0.9098	0.0008			
0.162	0.4555	0.2935			
0.301	0.3654	0.0644			
0.424	0.3556	0.0684			
0.533	0.4011	0.1319			
0.626	0.4851	0.1509			
0.702	0.6550	0.047			
0.763	0.7236	0.0394			
0.816	0.7800	0.036			
0.865	0.8420	0.023			
0.909	0.8992	0.0098			